BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

FIG. 1A

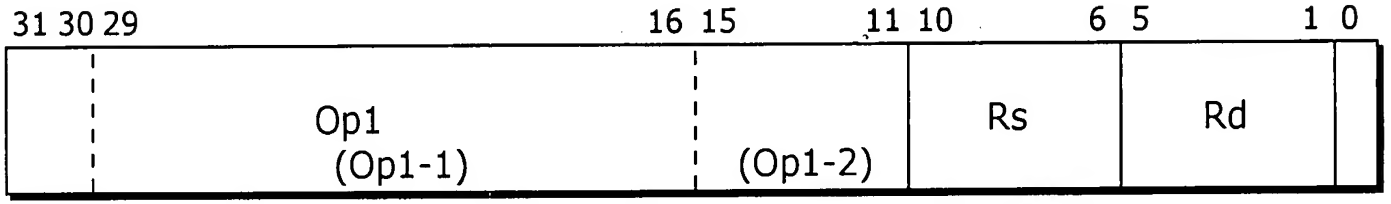


FIG. 1B

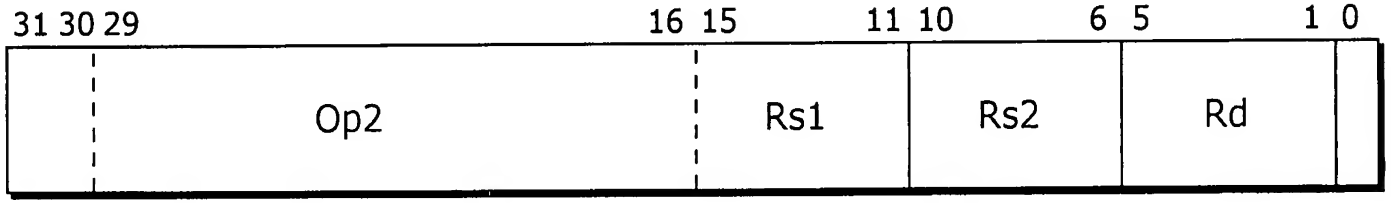


FIG. 1C

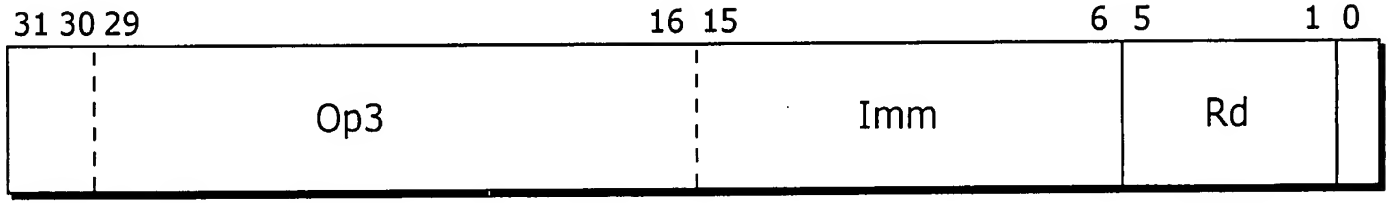


FIG. 1D

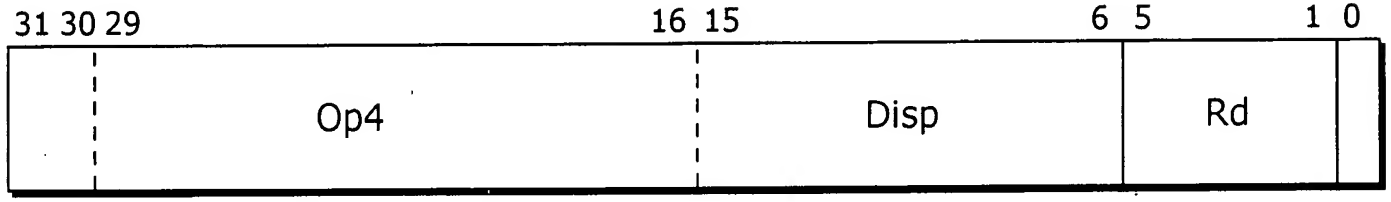


FIG. 2

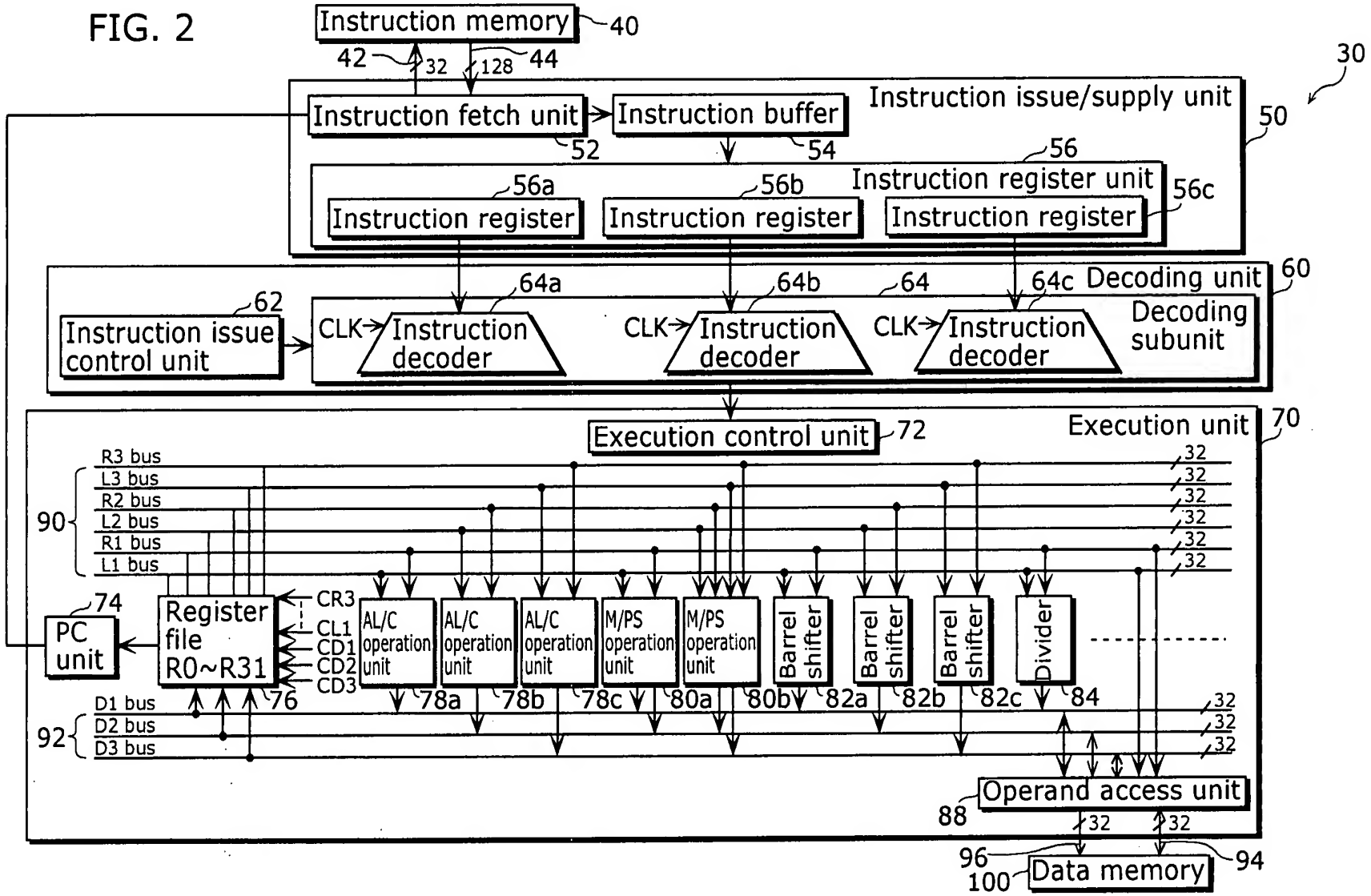

Instruction memory
40
42
32
128
44
30
Instruction issue/supply unit
50
Instruction fetch unit
52
Instruction buffer
54
56
Instruction register unit
56a
Instruction register
56b
Instruction register
56c
Instruction register
60
Decoding unit
62
Instruction issue control unit
64
64a
64b
64c
CLK
Instruction decoder
Decoding subunit
70
Execution unit
Execution control unit
72
R3 bus
L3 bus
R2 bus
L2 bus
R1 bus
L1 bus
90
74
PC unit
Register file R0~R31
76
CR3
CL1
CD1
CD2
CD3
AL/C operation unit
M/PS operation unit
Barrel shifter
Divider
78a
78b
78c
80a
80b
82a
82b
82c
84
D1 bus
D2 bus
D3 bus
92
Operand access unit
88
96
94
100
Data memory

FIG. 3

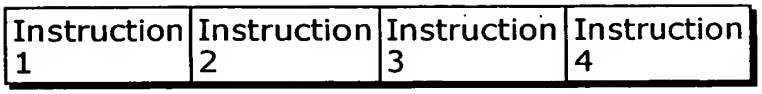


FIG. 4

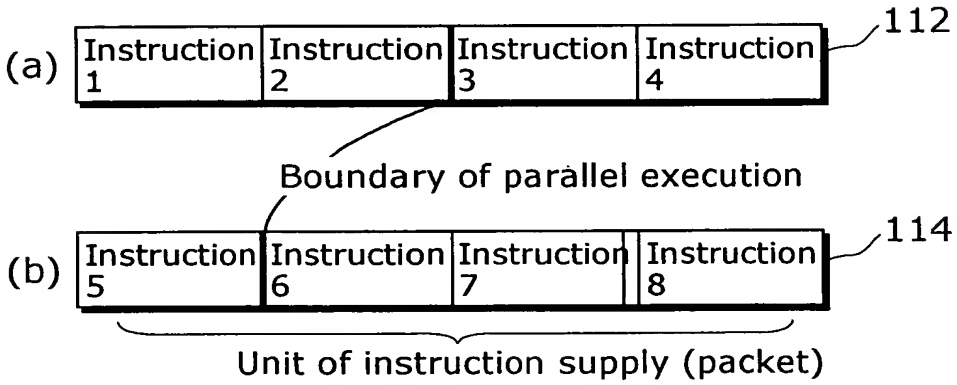


FIG. 5A

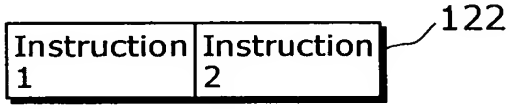


FIG. 5B

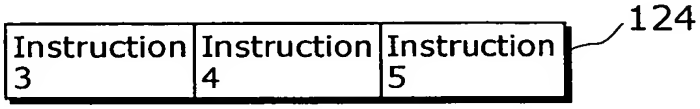


FIG. 5C

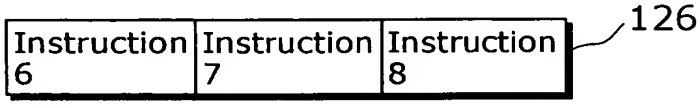


FIG. 6

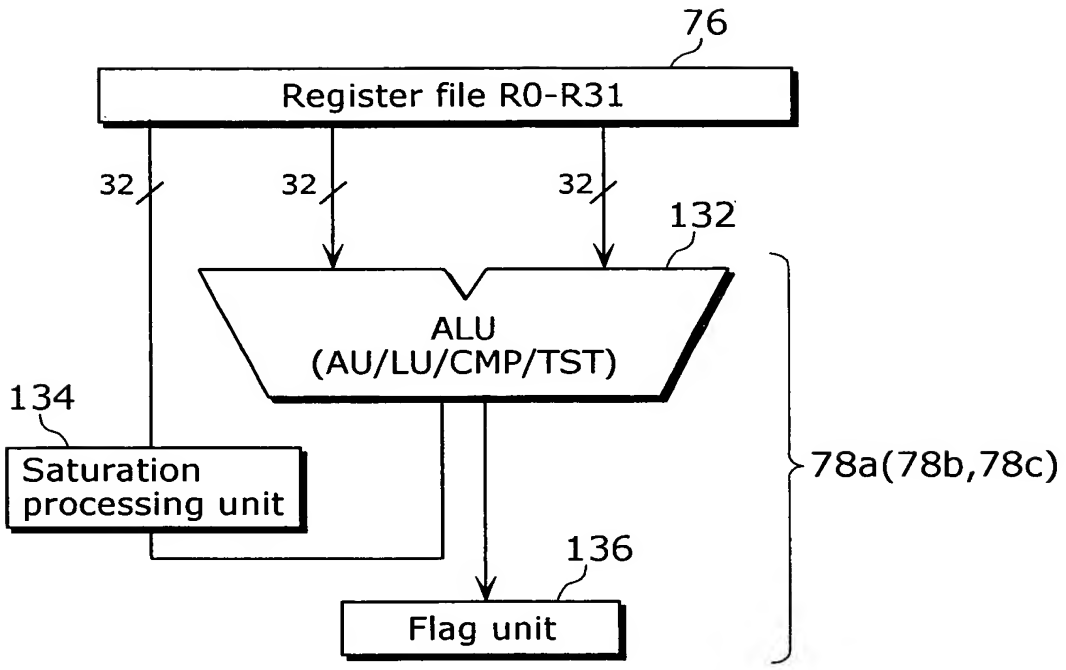

76
Register file R0-R31
32
32
32
132
ALU
(AU/LU/CMP/TST)
134
Saturation
processing unit
136
Flag unit
78a(78b,78c)

FIG. 7

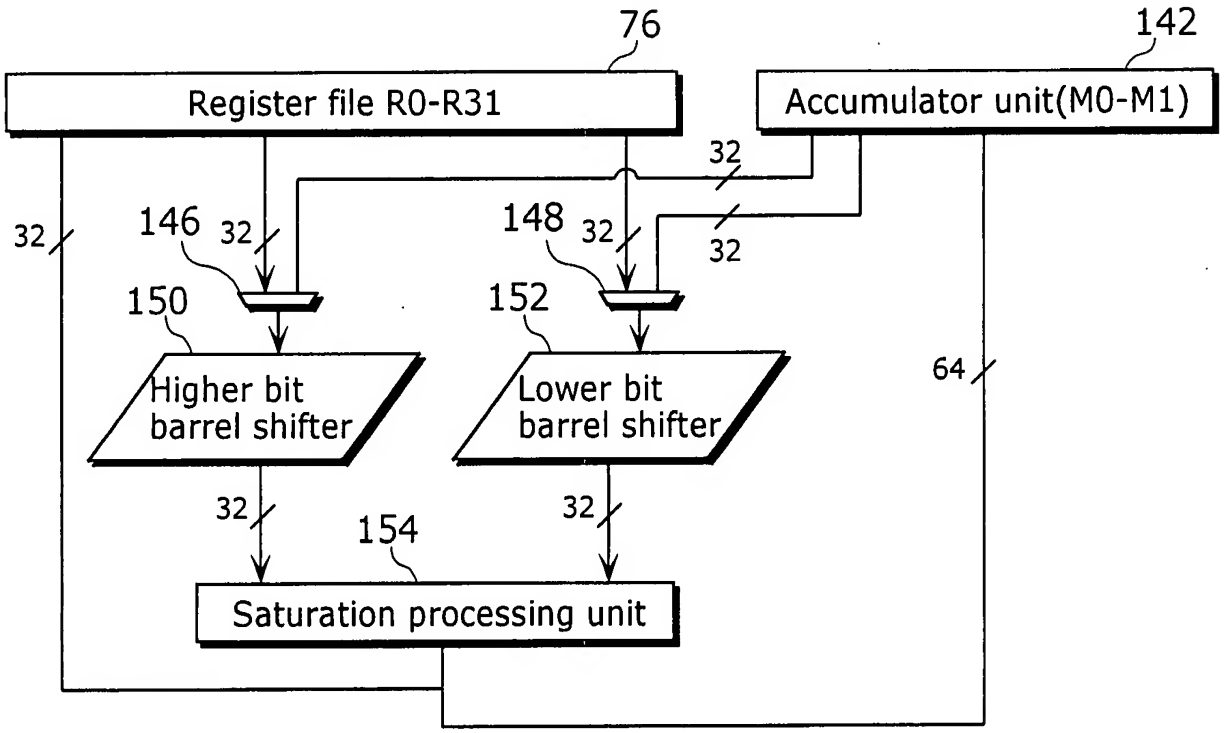

76
142
Register file R0-R31
Accumulator unit(M0-M1)
32
32
146
32
148
32
32
150
152
Higher bit barrel shifter
Lower bit barrel shifter
64
32
32
154
Saturation processing unit

FIG. 8

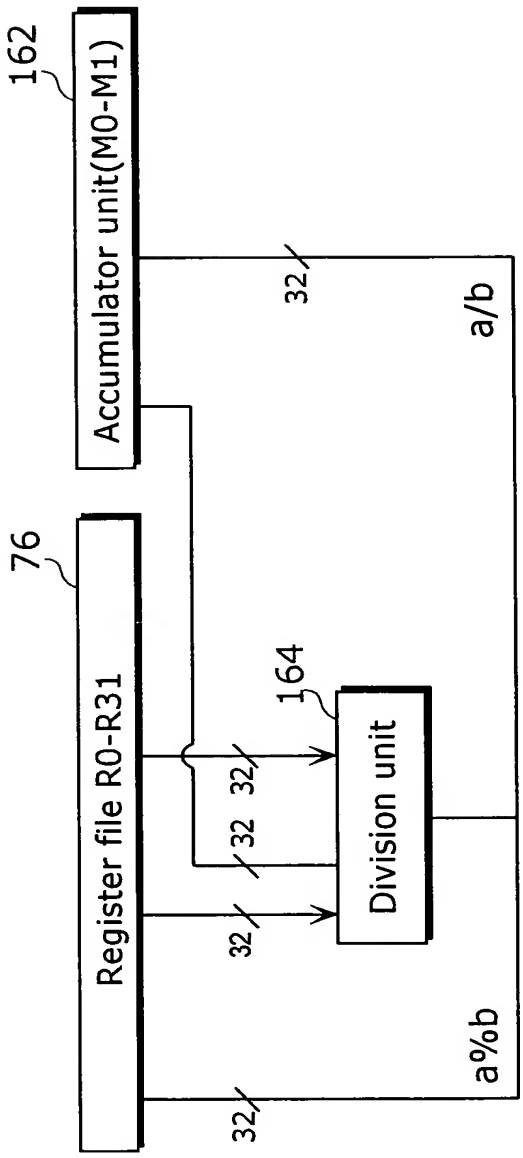

162
Accumulator unit(M0-M1)
32
a/b
76
Register file R0-R31
164
Division unit
32
32
32
a%b
32

FIG. 9

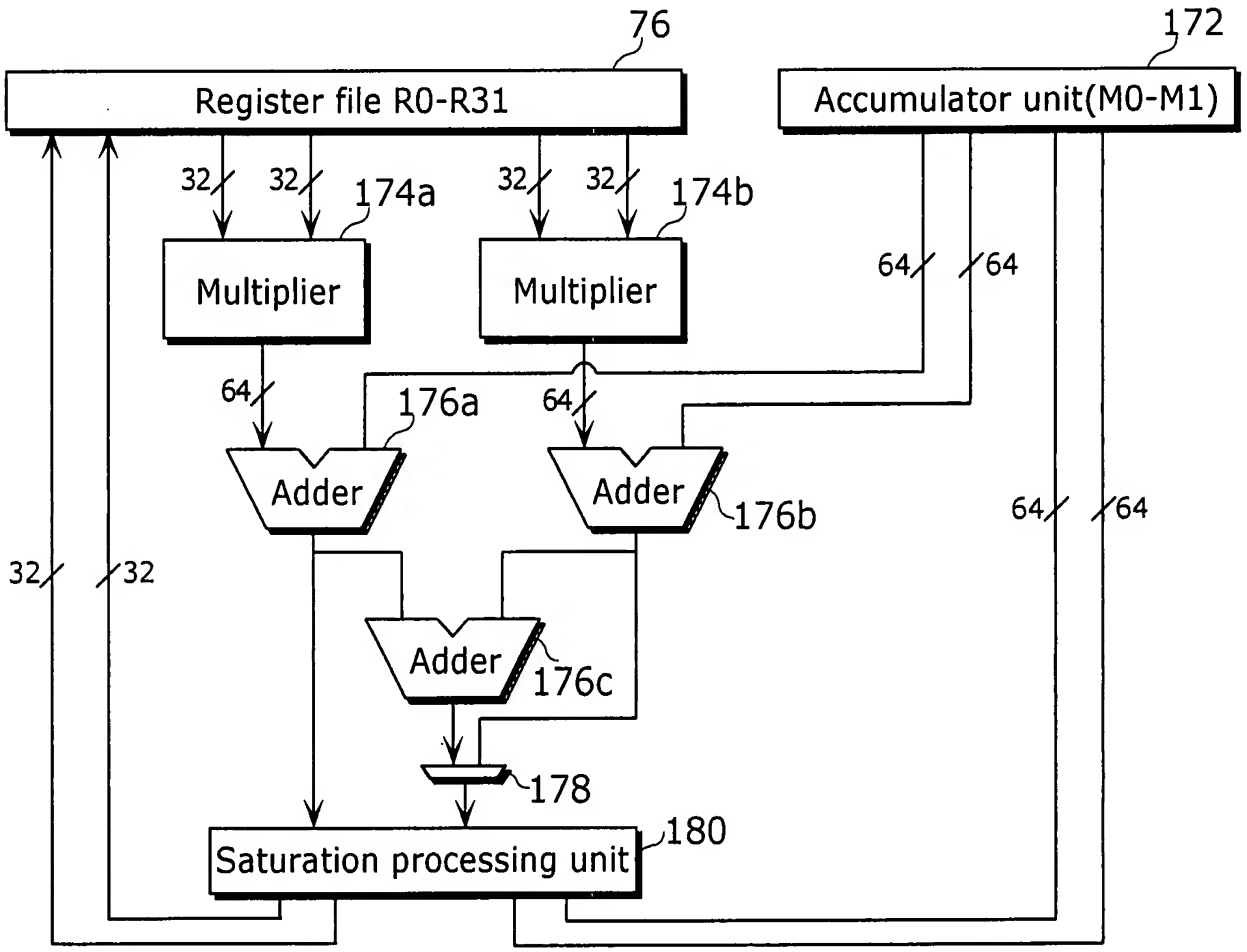

76
172
Register file R0-R31
Accumulator unit(M0-M1)
32
32
174a
32
32
174b
Multiplier
Multiplier
64
64
64
64
176a
Adder
Adder
176b
64
64
32
32
Adder
176c
178
180
Saturation processing unit

FIG. 10

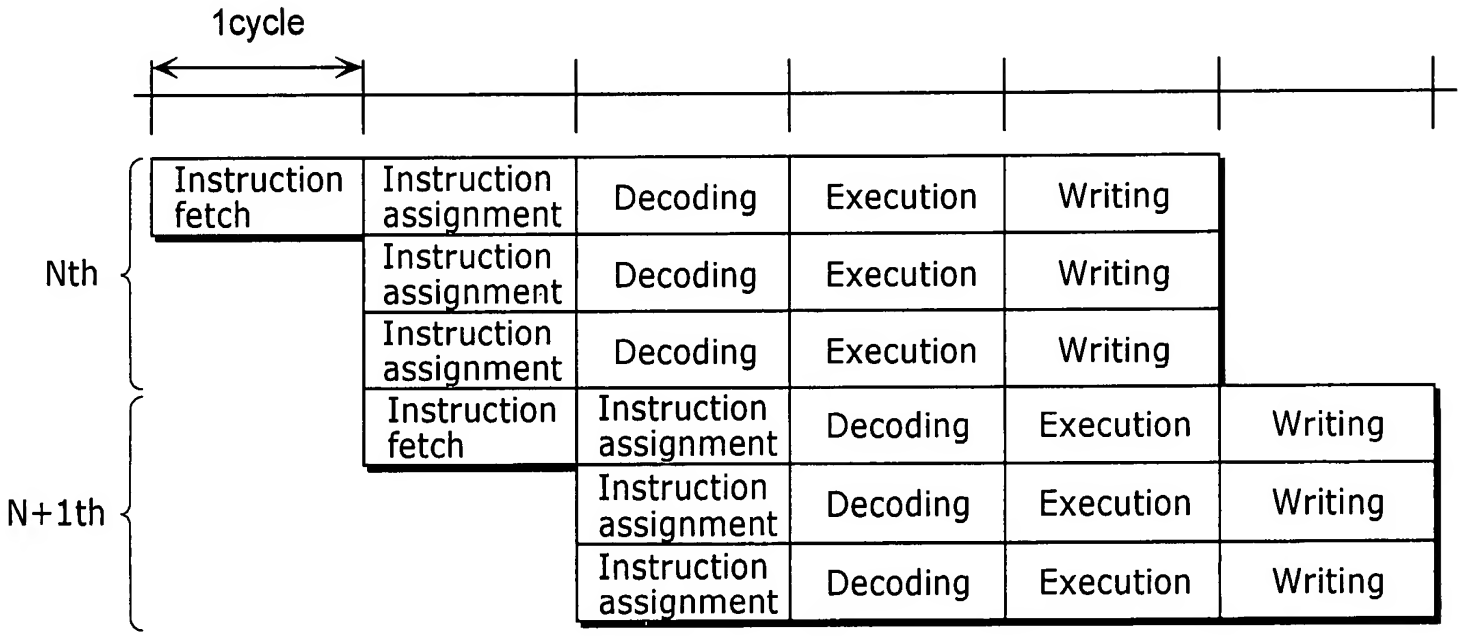

1cycle
Nth
N+1th
Instruction fetch
Instruction assignment
Decoding
Execution
Writing
Instruction fetch
Instruction assignment
Decoding
Execution
Writing

FIG. 11

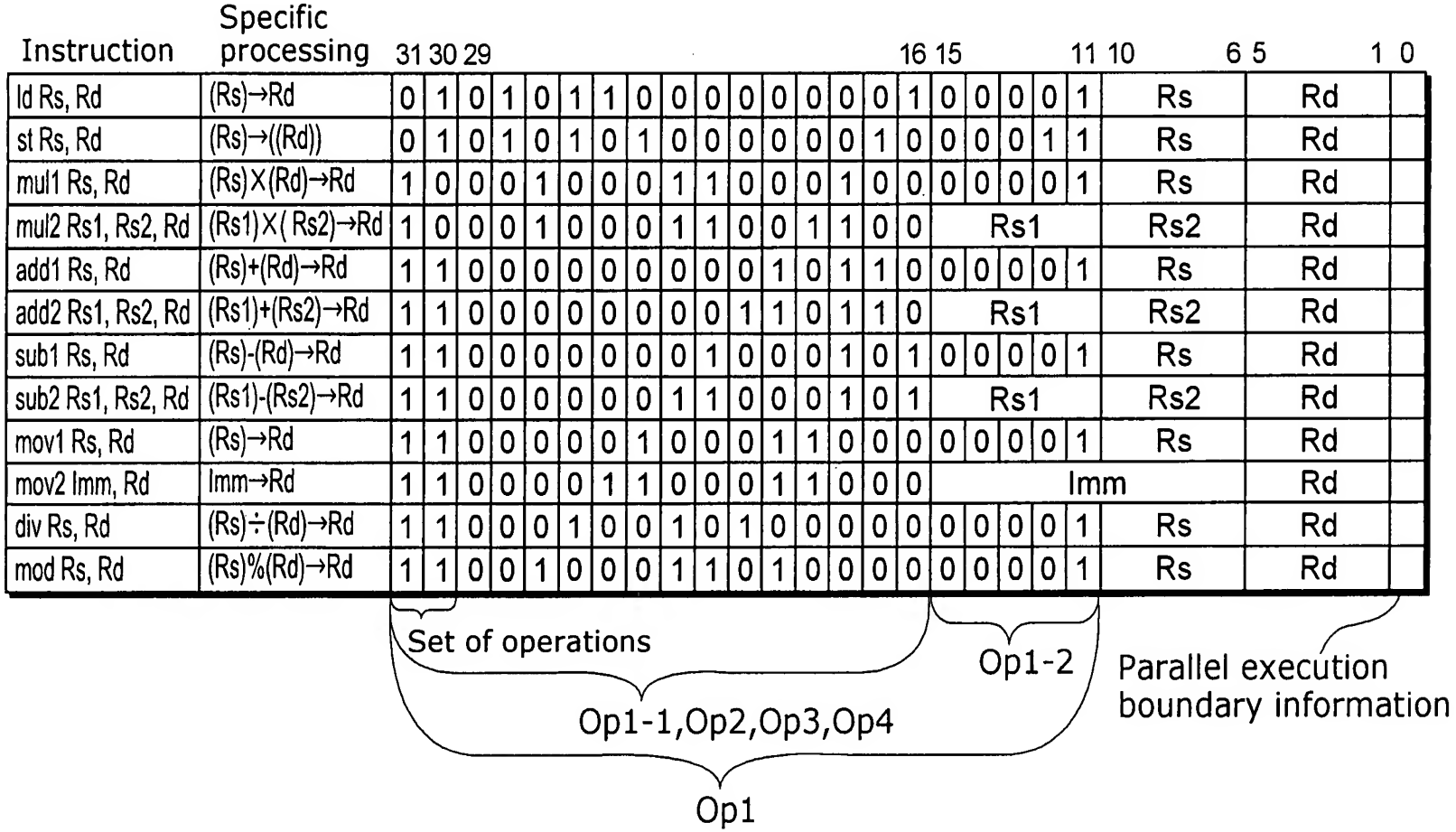

Instruction
Specific processing
31 30 29
16 15
11 10
6 5
1 0
ld Rs, Rd
(Rs)→Rd
st Rs, Rd
(Rs)→((Rd))
mul1 Rs, Rd
(Rs)X(Rd)→Rd
mul2 Rs1, Rs2, Rd
(Rs1)X(Rs2)→Rd
add1 Rs, Rd
(Rs)+(Rd)→Rd
add2 Rs1, Rs2, Rd
(Rs1)+(Rs2)→Rd
sub1 Rs, Rd
(Rs)-(Rd)→Rd
sub2 Rs1, Rs2, Rd
(Rs1)-(Rs2)→Rd
mov1 Rs, Rd
(Rs)→Rd
mov2 Imm, Rd
Imm→Rd
div Rs, Rd
(Rs)÷(Rd)→Rd
mod Rs, Rd
(Rs)%(Rd)→Rd
Rs
Rs1
Rs2
Imm
Rd
Set of operations
Op1-2
Parallel execution boundary information
Op1-1,Op2,Op3,Op4
Op1

FIG. 12

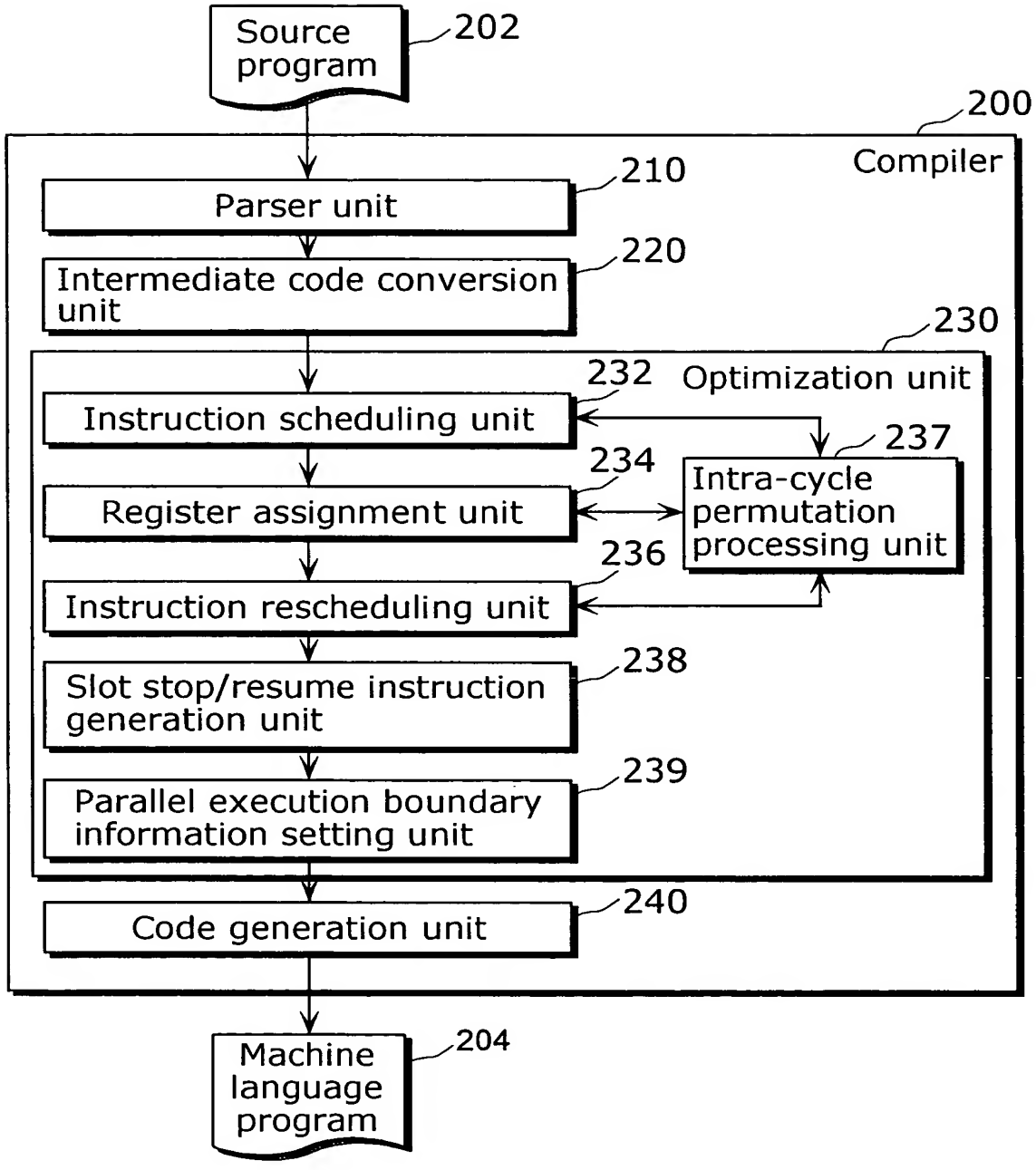

Source program
202
200
Compiler
210
Parser unit
220
Intermediate code conversion unit
230
Optimization unit
232
Instruction scheduling unit
237
234
Register assignment unit
Intra-cycle permutation processing unit
236
Instruction rescheduling unit
238
Slot stop/resume instruction generation unit
239
Parallel execution boundary information setting unit
240
Code generation unit
Machine language program
204

FIG. 13

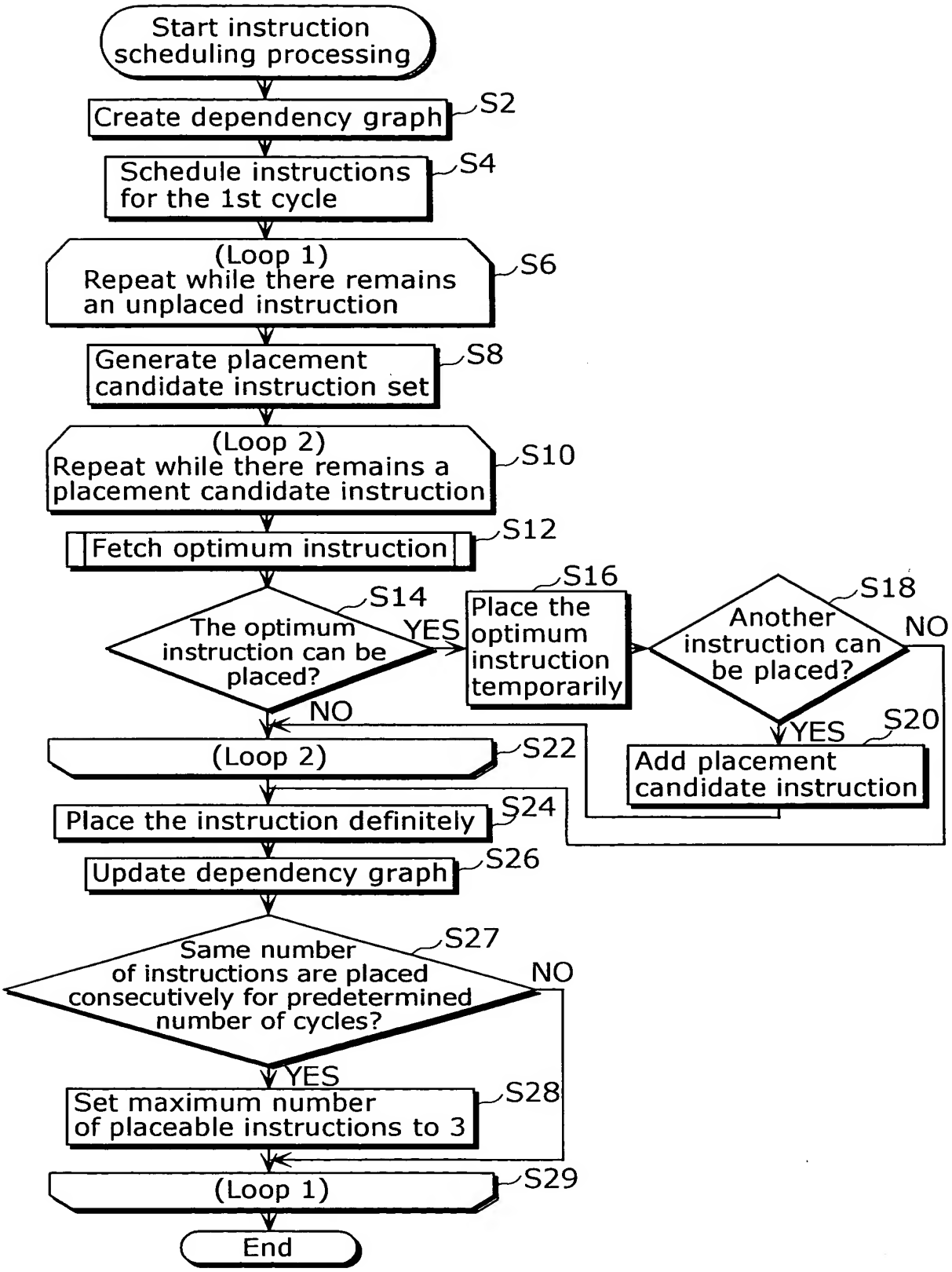

Start instruction scheduling processing
Create dependency graph
S2
Schedule instructions for the 1st cycle
S4
(Loop 1) Repeat while there remains an unplaced instruction
S6
Generate placement candidate instruction set
S8
(Loop 2) Repeat while there remains a placement candidate instruction
S10
Fetch optimum instruction
S12
The optimum instruction can be placed?
S14
YES
Place the optimum instruction temporarily
S16
Another instruction can be placed?
S18
NO
YES
Add placement candidate instruction
S20
NO
(Loop 2)
S22
Place the instruction definitely
S24
Update dependency graph
S26
Same number of instructions are placed consecutively for predetermined number of cycles?
S27
NO
YES
Set maximum number of placeable instructions to 3
S28
(Loop 1)
S29
End

FIG. 14A

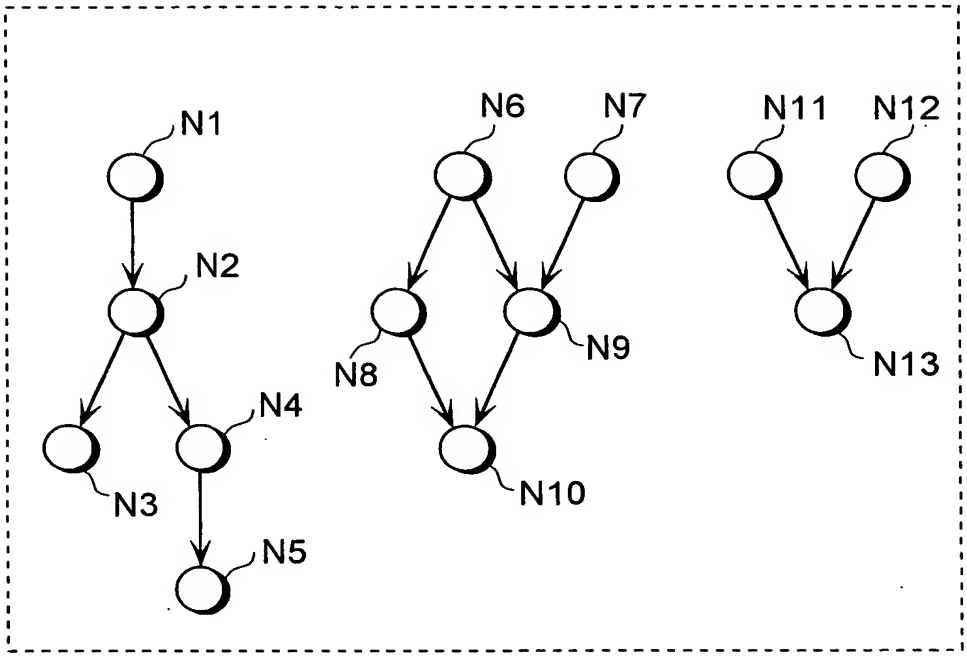


FIG. 14B

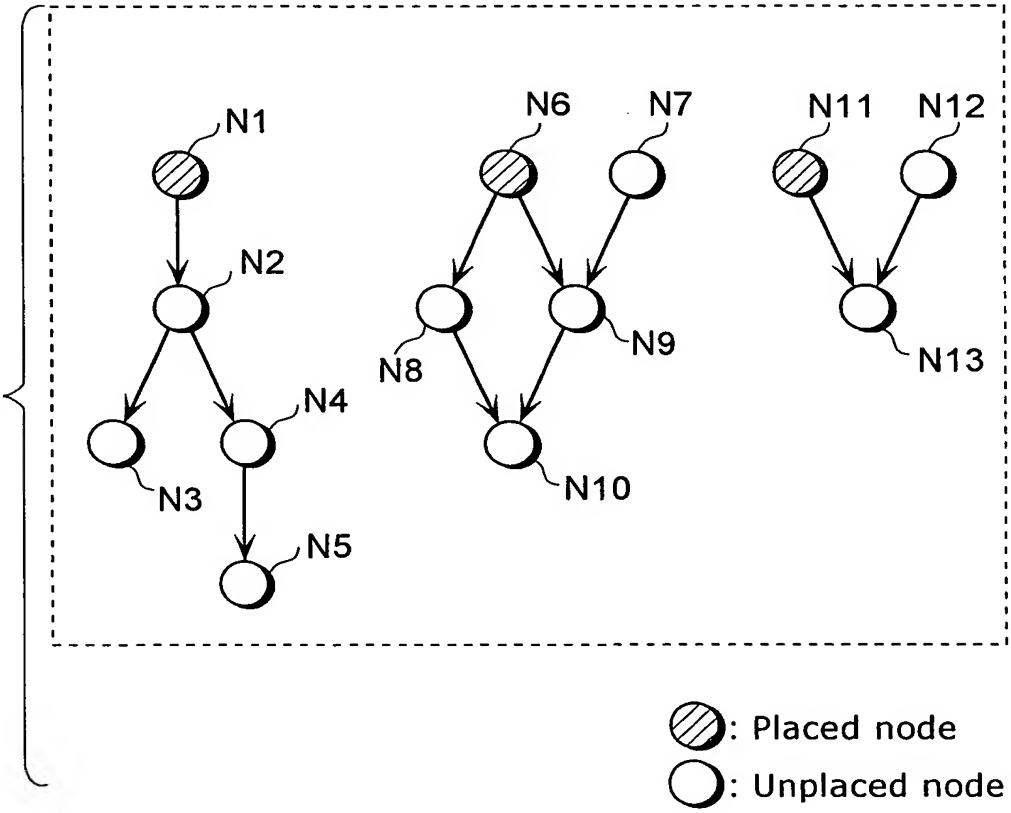


FIG. 15

Cycle	1st slot	2nd slot	3rd slot
1	ld Vr0, Vr1 (Node N1)	mul2 Vr2, Vr3, Vr4 (Node N11)	add1 Vr5, Vr6 (Node N6)

FIG. 16

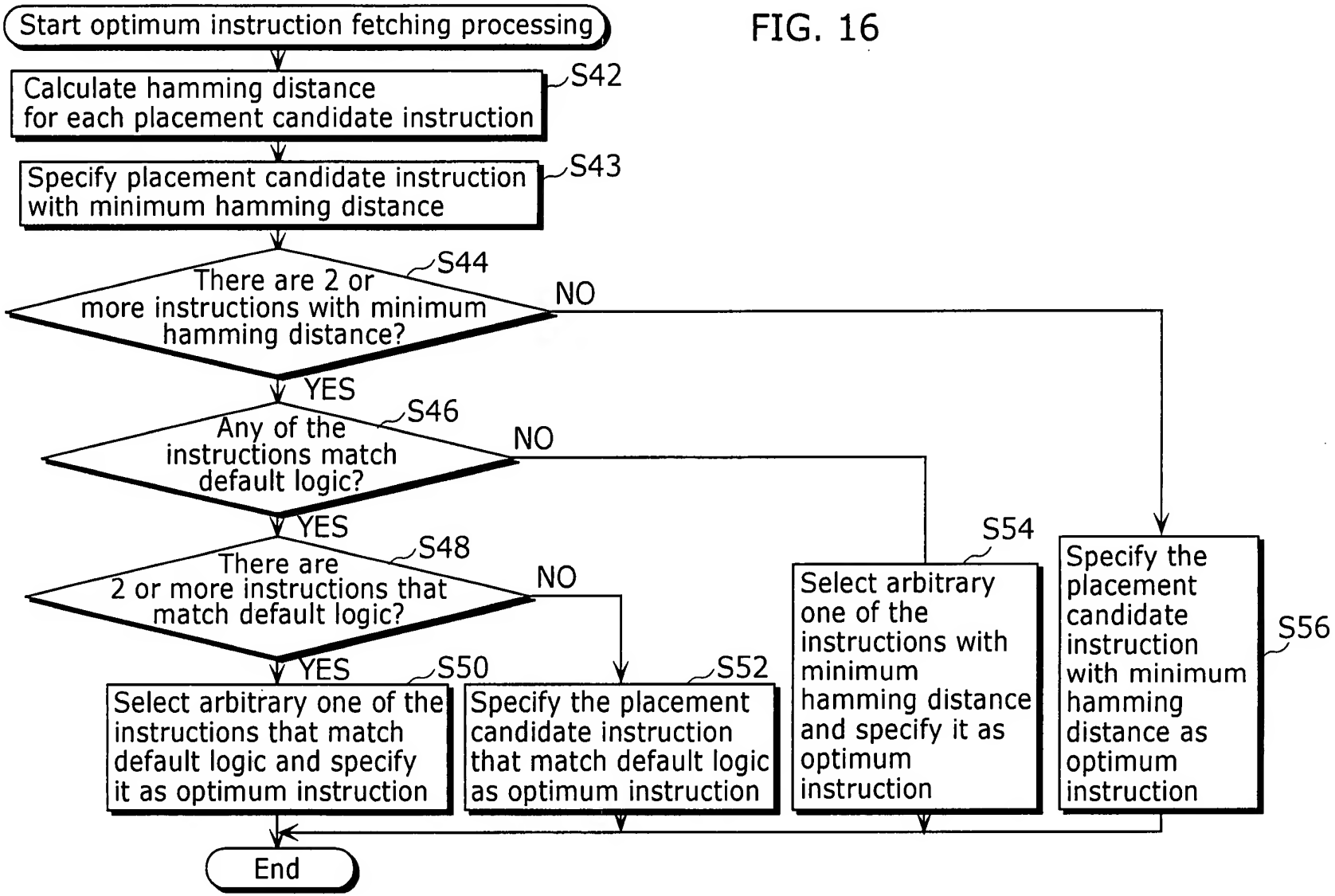

Start optimum instruction fetching processing
Calculate hamming distance for each placement candidate instruction
S42
Specify placement candidate instruction with minimum hamming distance
S43
There are 2 or more instructions with minimum hamming distance?
S44
NO
YES
Any of the instructions match default logic?
S46
NO
YES
There are 2 or more instructions that match default logic?
S48
NO
YES
Select arbitrary one of the instructions that match default logic and specify it as optimum instruction
S50
Specify the placement candidate instruction that match default logic as optimum instruction
S52
Select arbitrary one of the instructions with minimum hamming distance and specify it as optimum instruction
S54
Specify the placement candidate instruction with minimum hamming distance as optimum instruction
S56
End

FIG. 17A

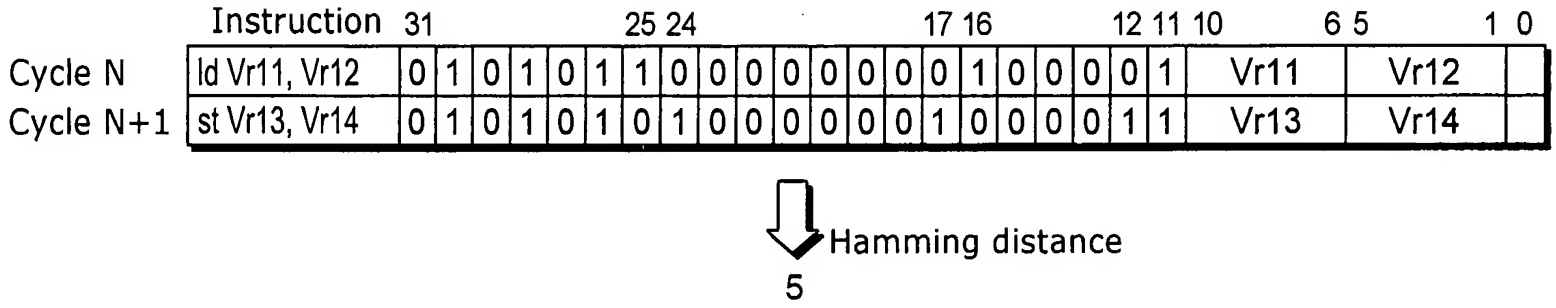


FIG. 17B

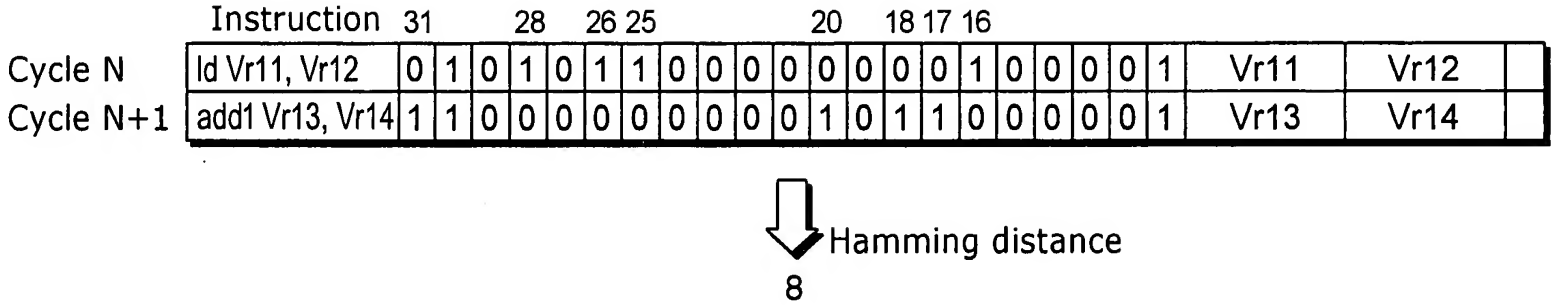


FIG. 18A

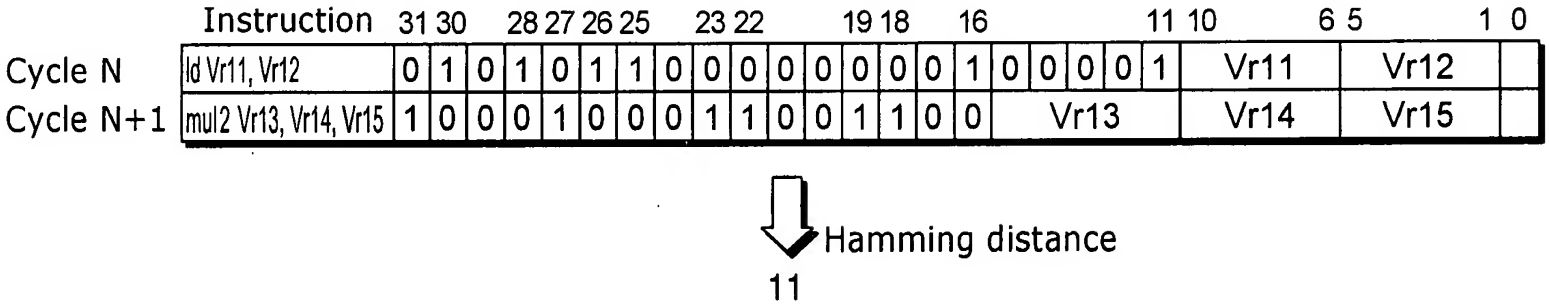


FIG. 18B

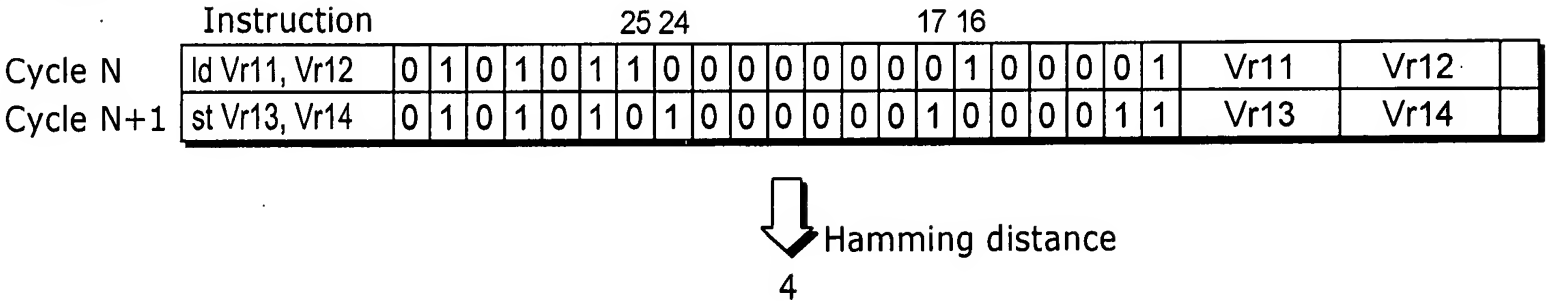


FIG. 19

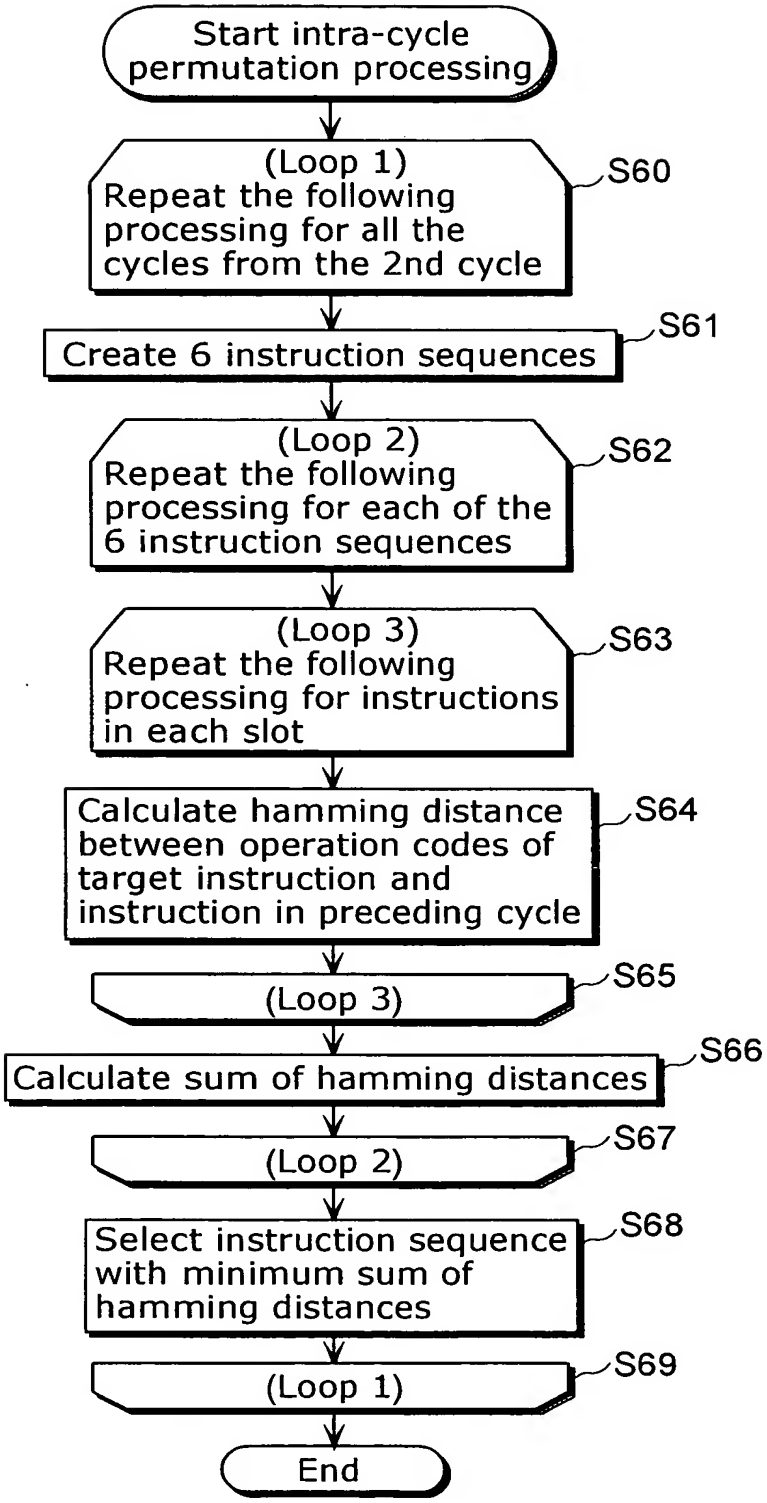

Start intra-cycle permutation processing
(Loop 1) Repeat the following processing for all the cycles from the 2nd cycle
S60
Create 6 instruction sequences
S61
(Loop 2) Repeat the following processing for each of the 6 instruction sequences
S62
(Loop 3) Repeat the following processing for instructions in each slot
S63
Calculate hamming distance between operation codes of target instruction and instruction in preceding cycle
S64
(Loop 3)
S65
Calculate sum of hamming distances
S66
(Loop 2)
S67
Select instruction sequence with minimum sum of hamming distances
S68
(Loop 1)
S69
End

FIG. 20A

FIG. 20B

FIG. 20C

FIG. 20D

FIG. 20E

FIG. 20F

FIG. 21

Cycle	1st slot			2nd slot			3rd slot		
⋮	⋮			⋮			⋮		
N	ld 01010110000000100001	Vr10	Vr11	sub1 11000000100010100001	Vr12	Vr13	add1 11000000001011000001	Vr14	Vr15
N+1	st 01010101000001000011	Vr16	Vr17	mul1 10001000110001000001	Vr18	Vr19	mod 11001000110100000001	Vr20	Vr21
⋮	⋮			⋮			⋮		

FIG. 22A

Cycle	1st slot			2nd slot			3rd slot		
N+1	st 01010101000000100011	Vr16	Vr17	mul1 10001000110001000001	Vr18	Vr19	mod 11001000110100000001	Vr20	Vr21

FIG. 22B

Cycle	1st slot			2nd slot			3rd slot		
N+1	st 01010101000000100011	Vr16	Vr17	mod 11001000110100000001	Vr20	Vr21	mul1 10001000110001000001	Vr18	Vr19

FIG. 22C

Cycle	1st slot			2nd slot			3rd slot		
N+1	mul1 10001000110001000001	Vr18	Vr19	st 01010101000000100011	Vr16	Vr17	mod 11001000110100000001	Vr20	Vr21

FIG. 22D

Cycle	1st slot			2nd slot			3rd slot		
N+1	mul1 10001000110001000001	Vr18	Vr19	mod 11001000110100000001	Vr20	Vr21	st 01010101000000100011	Vr16	Vr17

FIG. 22E

Cycle	1st slot			2nd slot			3rd slot		
N+1	mod 11001000110100000001	Vr20	Vr21	st 01010101000000100011	Vr16	Vr17	mul1 10001000110001000001	Vr18	Vr19

FIG. 22F

Cycle	1st slot			2nd slot			3rd slot		
N+1	mod 11001000110100000001	Vr20	Vr21	mul1 10001000110001000001	Vr18	Vr19	st 01010101000000100011	Vr16	Vr17

FIG. 23

Cycle	1st slot			2nd slot			3rd slot		
N	ld 01010110000000100001	Vr10	Vr11	sub1 11000000010001010001	Vr12	Vr13	add1 11000000001011000001	Vr14	Vr15
N+1	mul1 10001000110001000001	Vr18	Vr19	st 01010101000001000011	Vr16	Vr17	mod 11001000110100000001	Vr20	Vr21

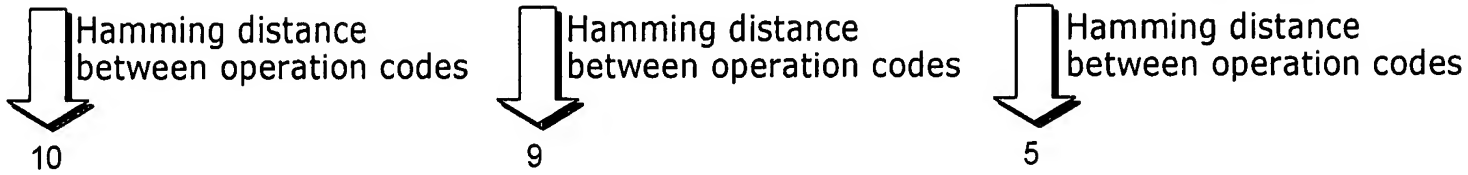


FIG. 24

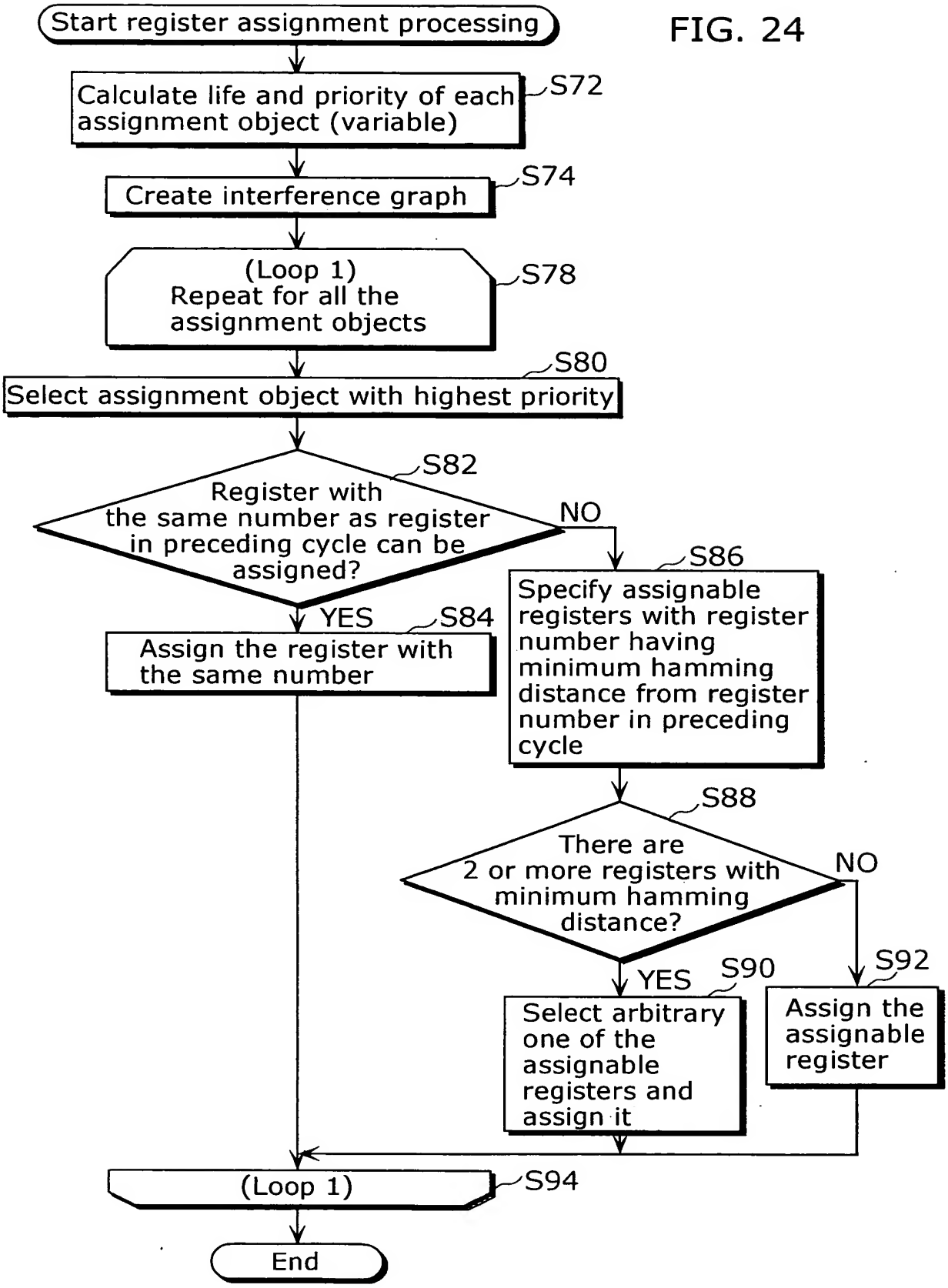

Start register assignment processing
Calculate life and priority of each assignment object (variable)
S72
Create interference graph
S74
(Loop 1)
Repeat for all the assignment objects
S78
Select assignment object with highest priority
S80
Register with the same number as register in preceding cycle can be assigned?
S82
NO
YES
S84
Assign the register with the same number
S86
Specify assignable registers with register number having minimum hamming distance from register number in preceding cycle
S88
There are 2 or more registers with minimum hamming distance?
NO
YES
S90
S92
Select arbitrary one of the assignable registers and assign it
Assign the assignable register
(Loop 1)
S94
End

FIG. 25

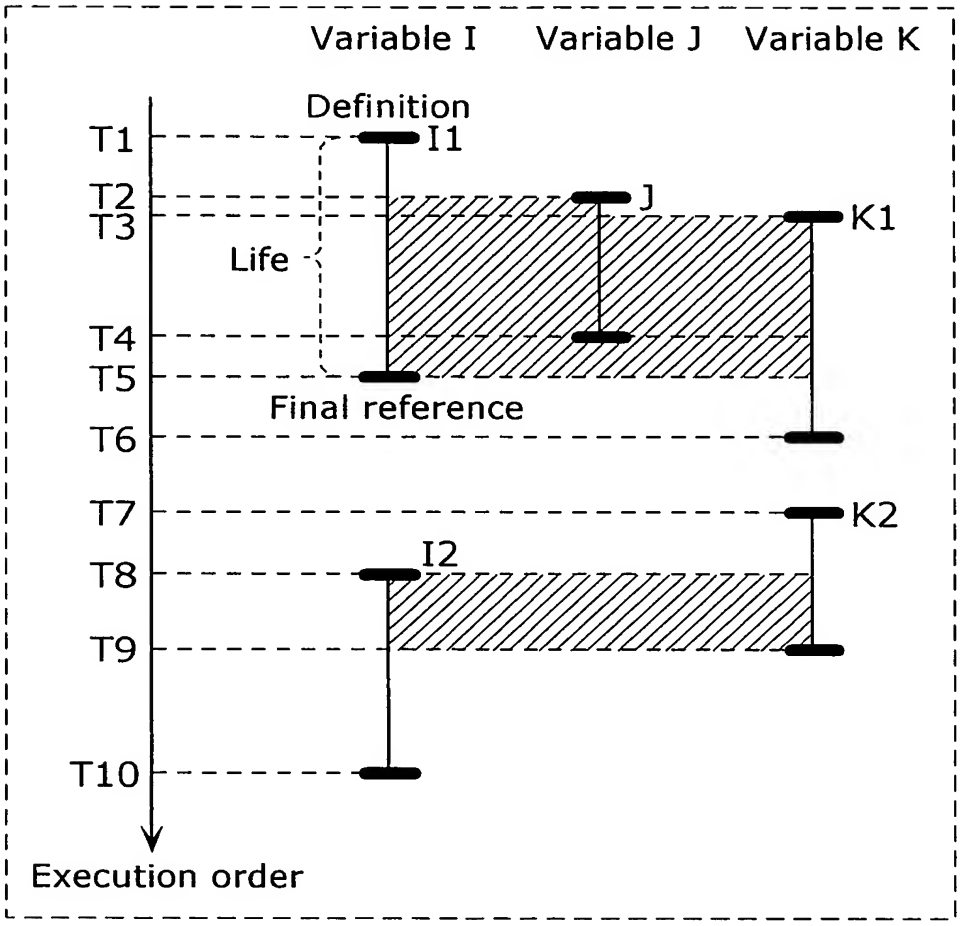

Variable I
Variable J
Variable K
Definition
T1
I1
T2
J
T3
K1
Life
T4
T5
Final reference
T6
T7
K2
I2
T8
T9
T10
Execution order

FIG. 26

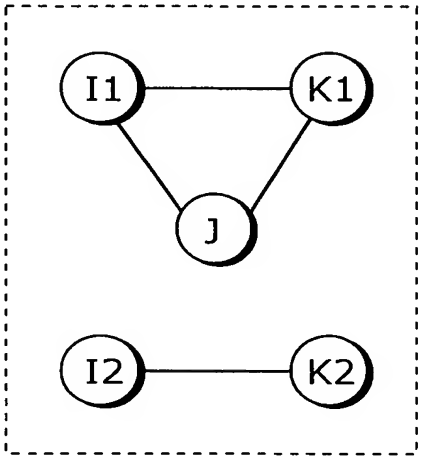

I1
K1
J
I2
K2

FIG. 27A

Cycle	1st slot	2nd slot	3rd slot
⋮	⋮	⋮	⋮
N	add1 R0, R2		
N+1	sub1 Vr5, Vr6		
⋮	⋮	⋮	⋮

FIG. 27B

N	add1 11000000000101100000 1	R0 00000	R2 00010
N+1	sub1 11000000010001010000 1	R0 00000	Vr6

FIG. 27C

N	add1 11000000000101100000 1	R0 00000	R2 00010
N+1	sub1 11000000010001010000 1	R1 00001	Vr6

FIG. 28

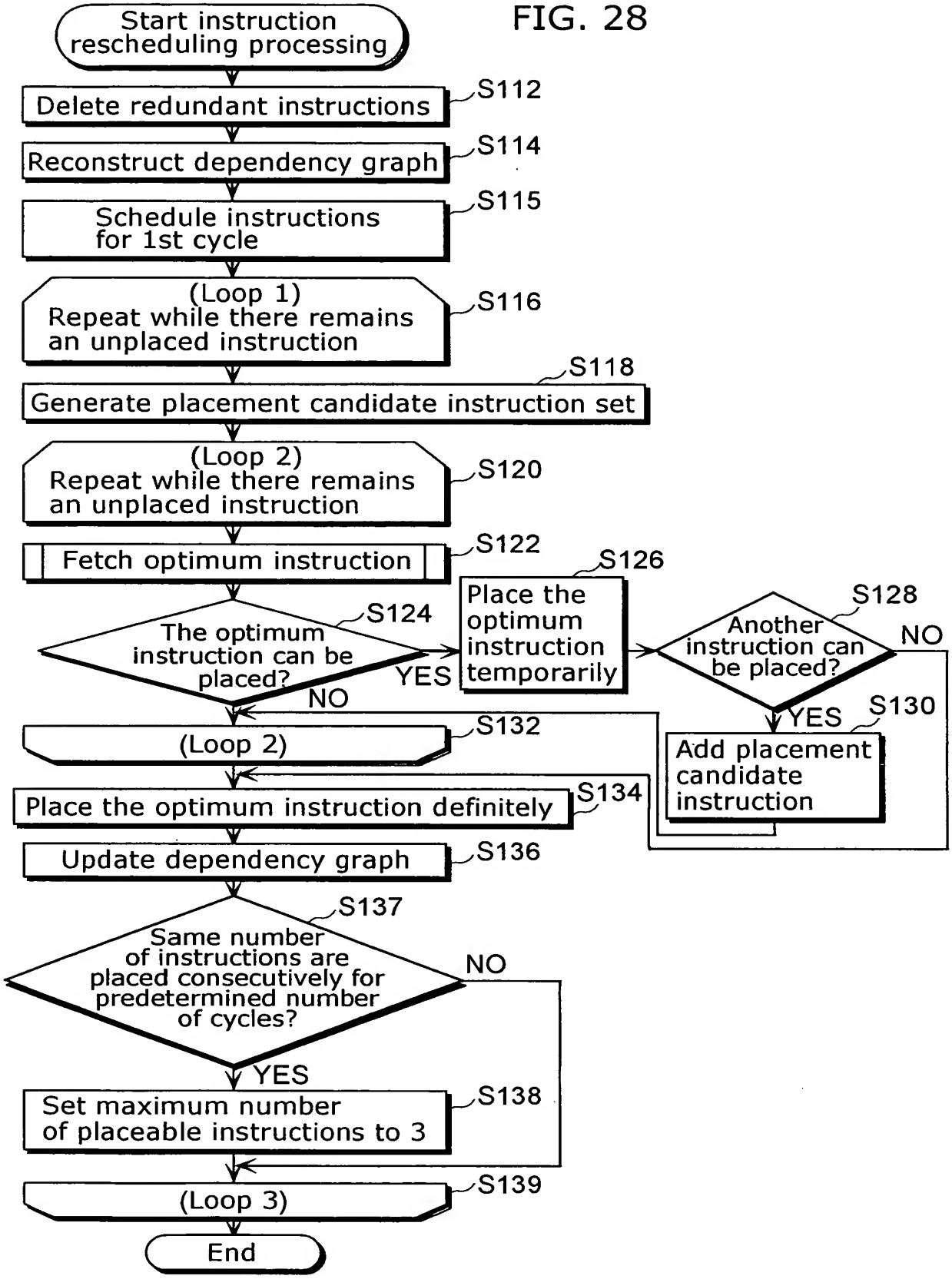

Start instruction rescheduling processing
Delete redundant instructions
S112
Reconstruct dependency graph
S114
Schedule instructions for 1st cycle
S115
(Loop 1) Repeat while there remains an unplaced instruction
S116
Generate placement candidate instruction set
S118
(Loop 2) Repeat while there remains an unplaced instruction
S120
Fetch optimum instruction
S122
The optimum instruction can be placed?
S124
YES
NO
Place the optimum instruction temporarily
S126
Another instruction can be placed?
S128
NO
YES
Add placement candidate instruction
S130
(Loop 2)
S132
Place the optimum instruction definitely
S134
Update dependency graph
S136
Same number of instructions are placed consecutively for predetermined number of cycles?
S137
NO
YES
Set maximum number of placeable instructions to 3
S138
(Loop 3)
S139
End

FIG. 29

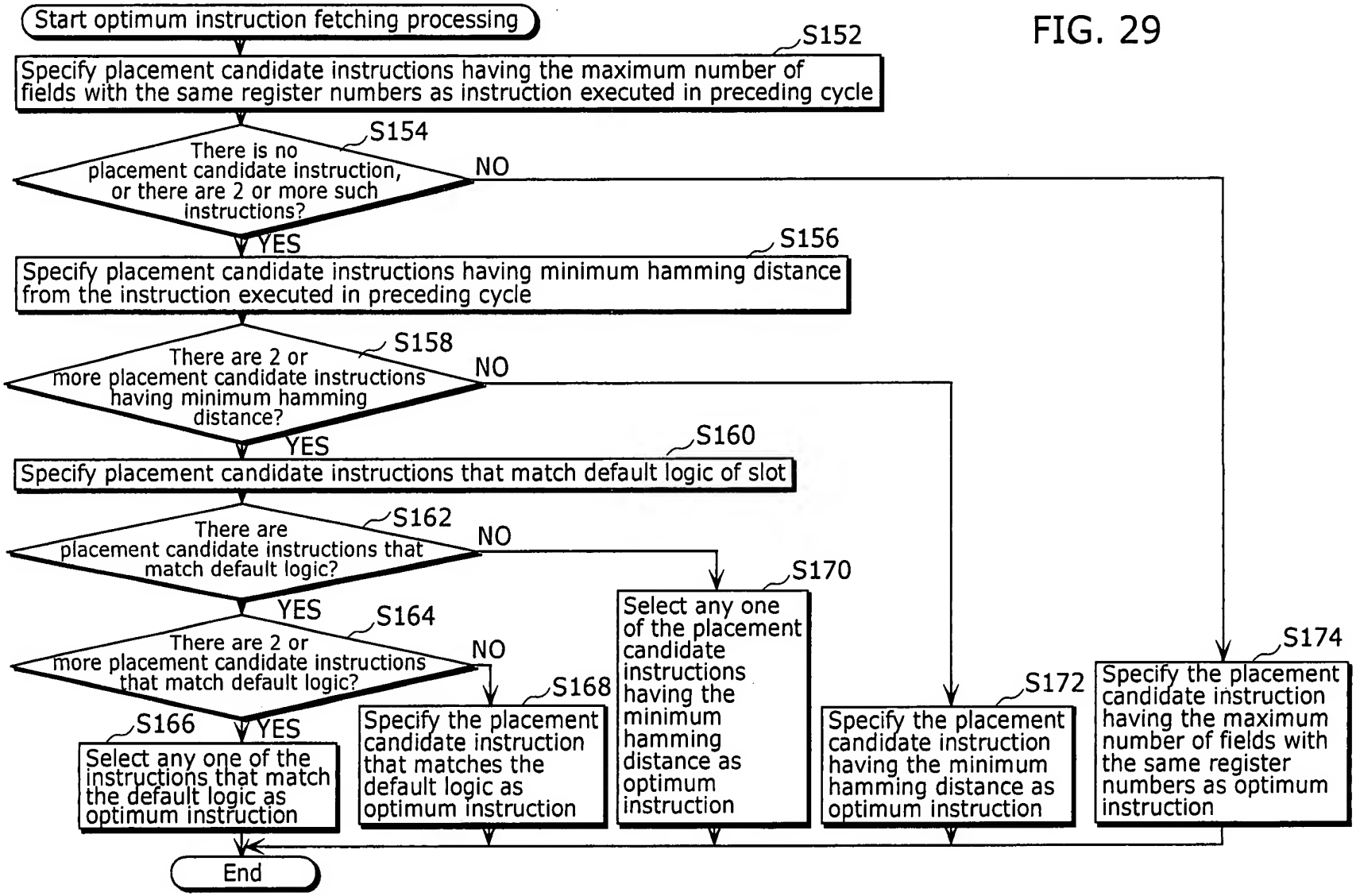

Start optimum instruction fetching processing
S152
Specify placement candidate instructions having the maximum number of fields with the same register numbers as instruction executed in preceding cycle
S154
There is no placement candidate instruction, or there are 2 or more such instructions?
NO
YES
S156
Specify placement candidate instructions having minimum hamming distance from the instruction executed in preceding cycle
S158
There are 2 or more placement candidate instructions having minimum hamming distance?
NO
YES
S160
Specify placement candidate instructions that match default logic of slot
S162
There are placement candidate instructions that match default logic?
NO
YES
S164
There are 2 or more placement candidate instructions that match default logic?
NO
YES
S166
Select any one of the instructions that match the default logic as optimum instruction
S168
Specify the placement candidate instruction that matches the default logic as optimum instruction
S170
Select any one of the placement candidate instructions having the minimum hamming distance as optimum instruction
S172
Specify the placement candidate instruction having the minimum hamming distance as optimum instruction
S174
Specify the placement candidate instruction having the maximum number of fields with the same register numbers as optimum instruction
End

FIG. 30A

Cycle	1st slot		
N	add1 11000000000101100000 1	R0 00000	R2 00010
N+1	sub1 11000000010001010000 1	R0 00000	R1 00001

Number of register fields having the same register numbers: 1

FIG. 30B

N	add1 11000000000101100000 1	R0 00000	R2 00010
N+1	div 11000100101000000000 1	R0 00000	R2 00010

Number of register fields having the same register numbers: 2

FIG. 31A

Cycle	1st slot		
N	mul1 1000100011000010000001	R3 00011	R10 01010
N+1	add1 1100000000001011000001	R2 00010	R4 00100

10 Hamming distance between instructions

FIG. 31B

N	mul1 1000100011000010000001		R3 00011	R10 01010
N+1	sub2 1100000011000101	R11 01011	R0 00000	R2 00010

8 Hamming distance between instructions

FIG. 32A

Cycle	1st slot		
N	st 01010101000000100011	R1 00001	R13 01101
N+1	ld 01010110000000100001	R30 11110	R18 10010

⇓⇓ Default logic of 1st slot

Match

FIG. 32B

N	st 01010101000000100011	R1 00001	R13 01101
N+1	sub1 11000000010010100001	R8 01000	R2 00010

⇓⇓ Default logic of 1st slot

Does not match

FIG. 33

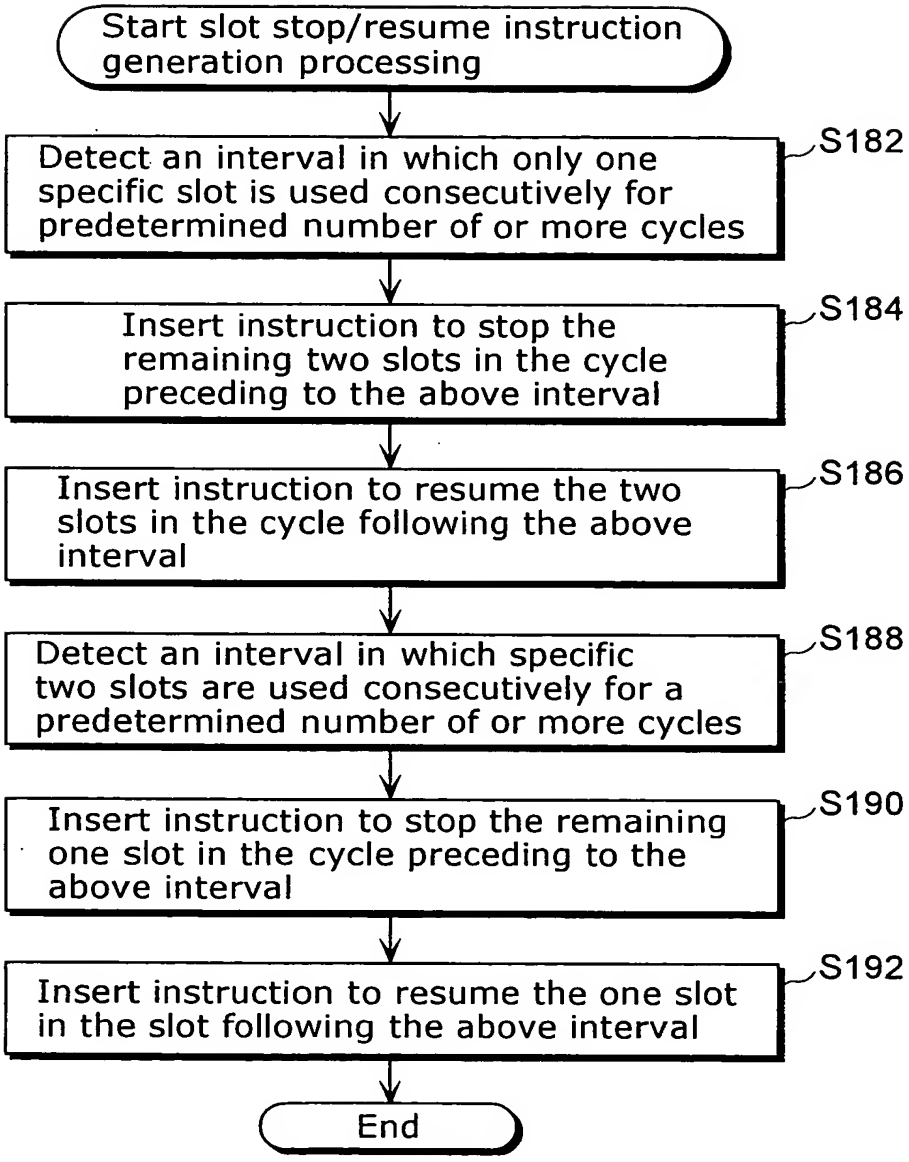

Start slot stop/resume instruction generation processing
Detect an interval in which only one specific slot is used consecutively for predetermined number of or more cycles
S182
Insert instruction to stop the remaining two slots in the cycle preceding to the above interval
S184
Insert instruction to resume the two slots in the cycle following the above interval
S186
Detect an interval in which specific two slots are used consecutively for a predetermined number of or more cycles
S188
Insert instruction to stop the remaining one slot in the cycle preceding to the above interval
S190
Insert instruction to resume the one slot in the slot following the above interval
S192
End

FIG. 34

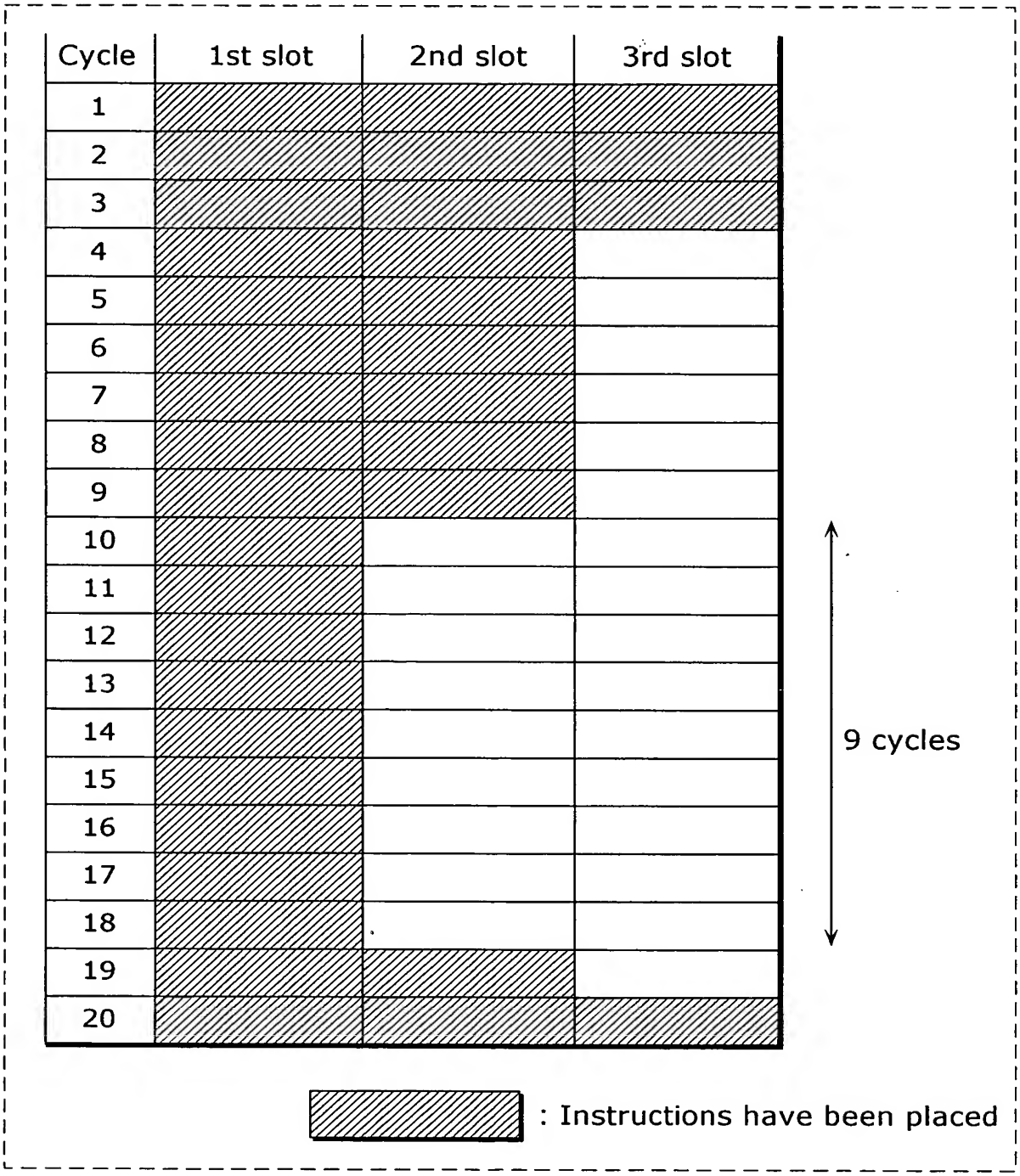

Cycle
1st slot
2nd slot
3rd slot
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
9 cycles
: Instructions have been placed

FIG. 35

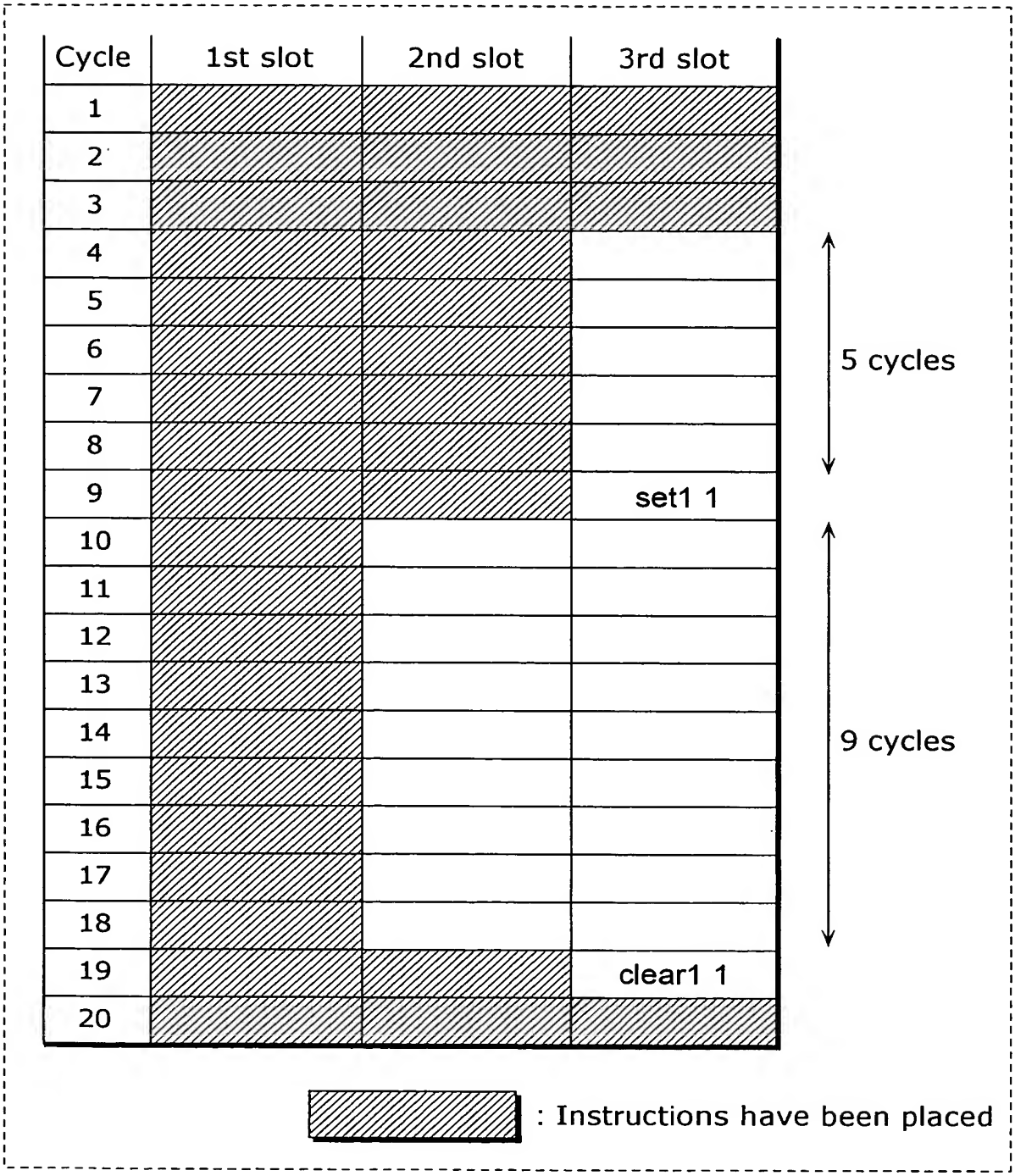

Cycle
1st slot
2nd slot
3rd slot
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
set1 1
clear1 1
5 cycles
9 cycles
: Instructions have been placed

FIG. 36

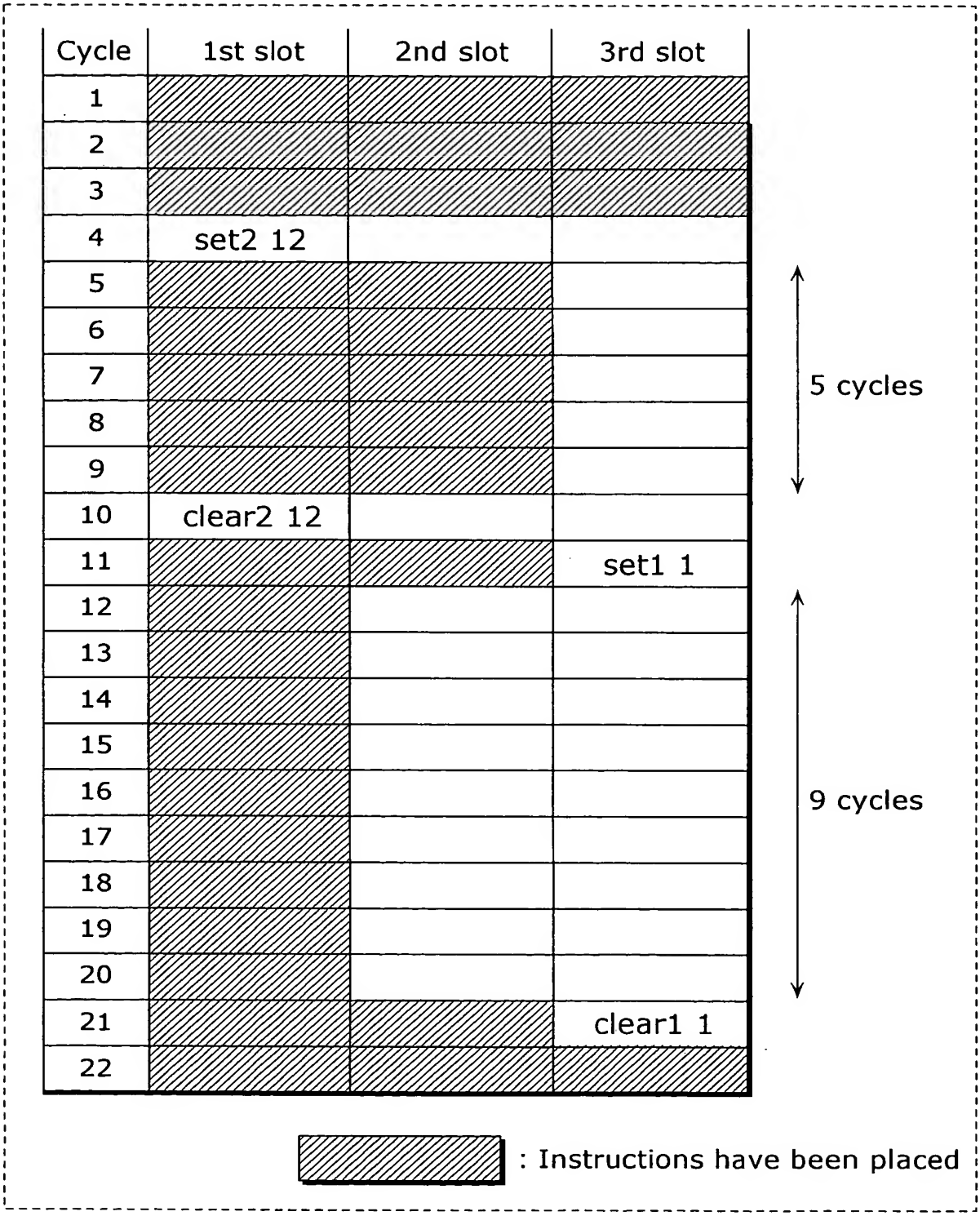

Cycle
1st slot
2nd slot
3rd slot
1
2
3
4
set2 12
5
6
7
8
9
10
clear2 12
11
set1 1
12
13
14
15
16
17
18
19
20
21
clear1 1
22
5 cycles
9 cycles
: Instructions have been placed

FIG. 37

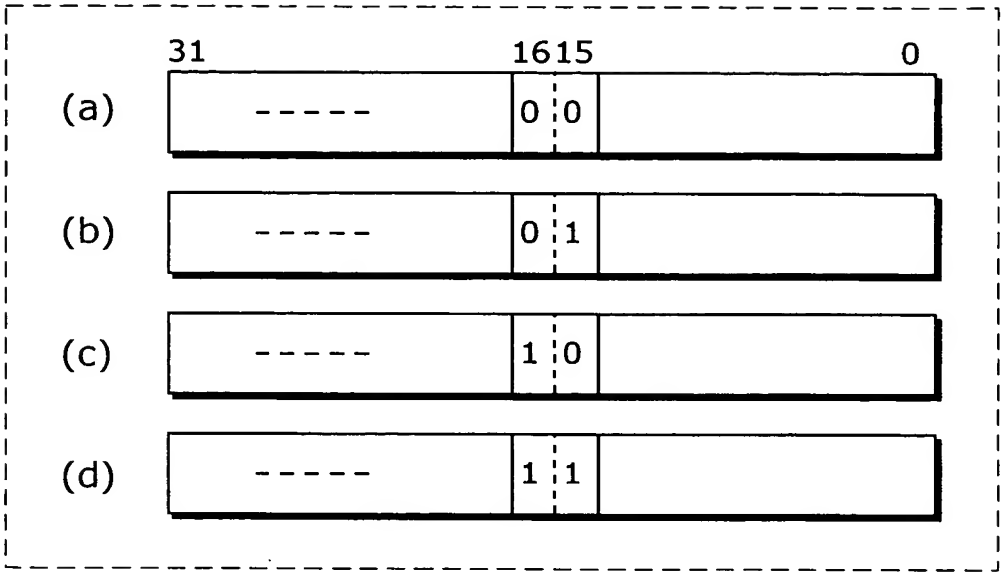

31
16 15
0
(a)
-----
0 0
(b)
-----
0 1
(c)
-----
1 0
(d)
-----
1 1

FIG. 38

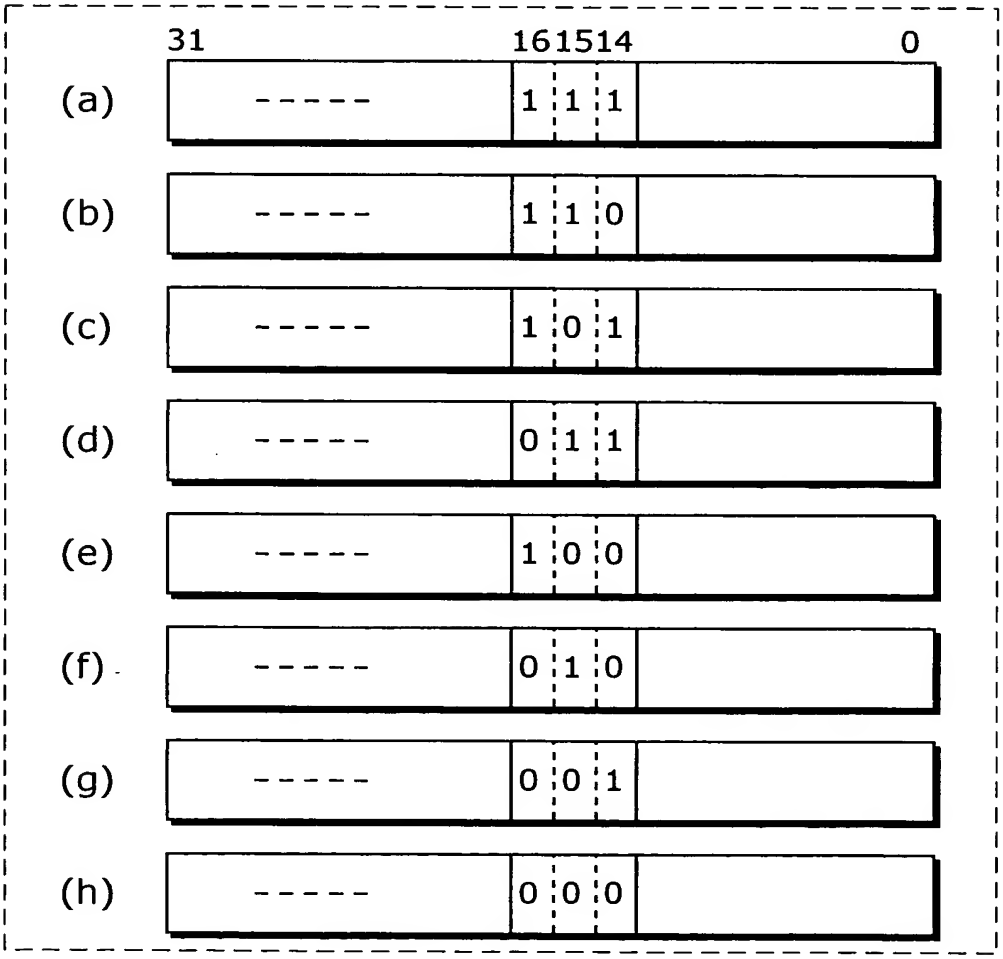

31
16 15 14
0
(a)
-----
1 1 1
(b)
-----
1 1 0
(c)
-----
1 0 1
(d)
-----
0 1 1
(e)
-----
1 0 0
(f)
-----
0 1 0
(g)
-----
0 0 1
(h)
-----
0 0 0

FIG. 39

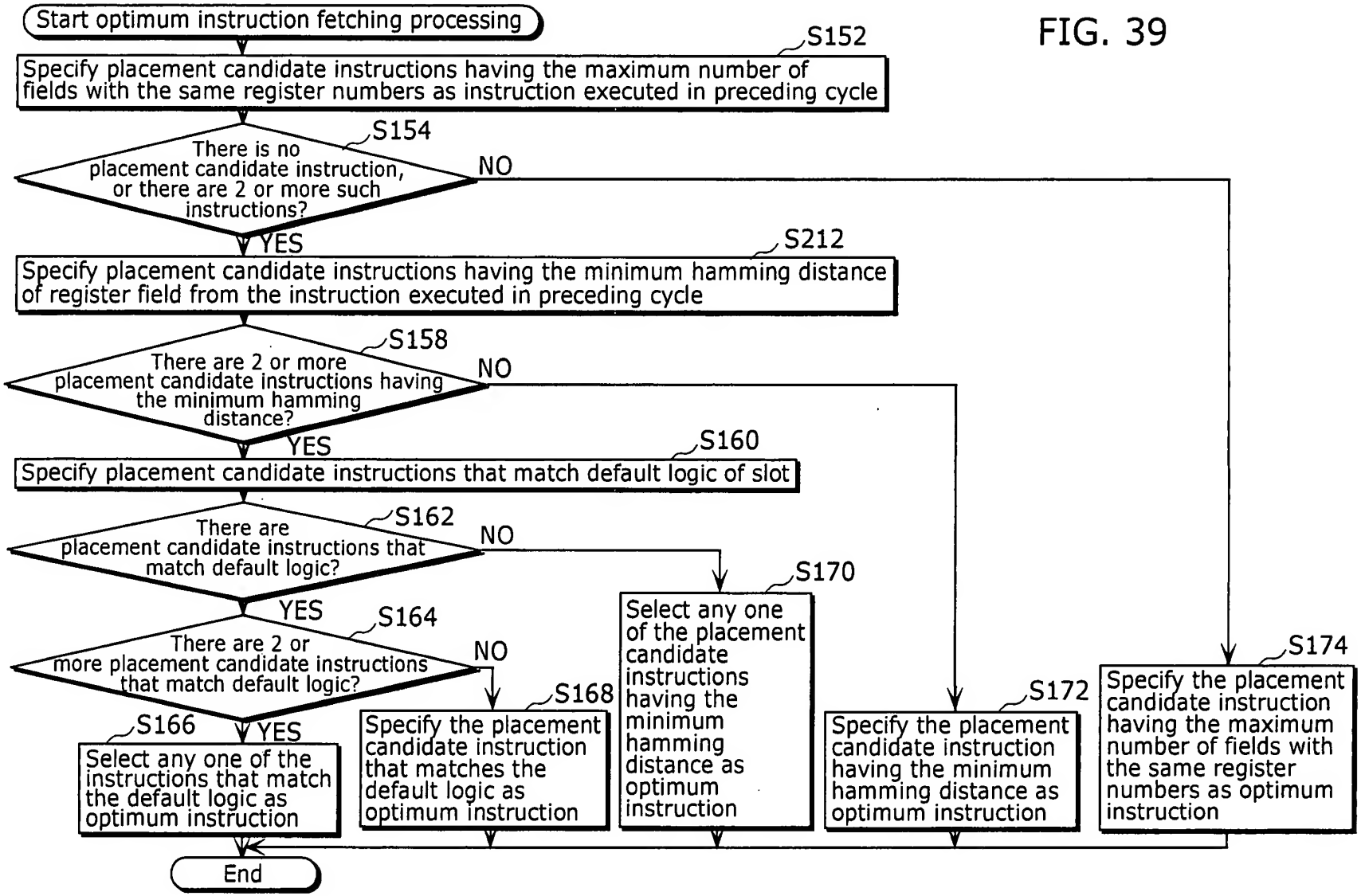

Start optimum instruction fetching processing
S152
Specify placement candidate instructions having the maximum number of fields with the same register numbers as instruction executed in preceding cycle
S154
There is no placement candidate instruction, or there are 2 or more such instructions?
NO
YES
S212
Specify placement candidate instructions having the minimum hamming distance of register field from the instruction executed in preceding cycle
S158
There are 2 or more placement candidate instructions having the minimum hamming distance?
NO
YES
S160
Specify placement candidate instructions that match default logic of slot
S162
There are placement candidate instructions that match default logic?
NO
YES
S164
There are 2 or more placement candidate instructions that match default logic?
NO
YES
S166
Select any one of the instructions that match the default logic as optimum instruction
S168
Specify the placement candidate instruction that matches the default logic as optimum instruction
S170
Select any one of the placement candidate instructions having the minimum hamming distance as optimum instruction
S172
Specify the placement candidate instruction having the minimum hamming distance as optimum instruction
S174
Specify the placement candidate instruction having the maximum number of fields with the same register numbers as optimum instruction
End

FIG. 40A

Cycle	1st slot		
N	add1 11000000000101100000 1	R0 00000	R2 00010
N+1	sub1 11000000010001010000 1	R3 00011	R1 00001

Hamming distance between register fields
4

FIG. 40B

N	add1 11000000000101100000 1	R0 00000	R2 00010
N+1	div 11000100101000000000 1	R7 00111	R1 00001

Hamming distance between register fields
5

FIG. 41

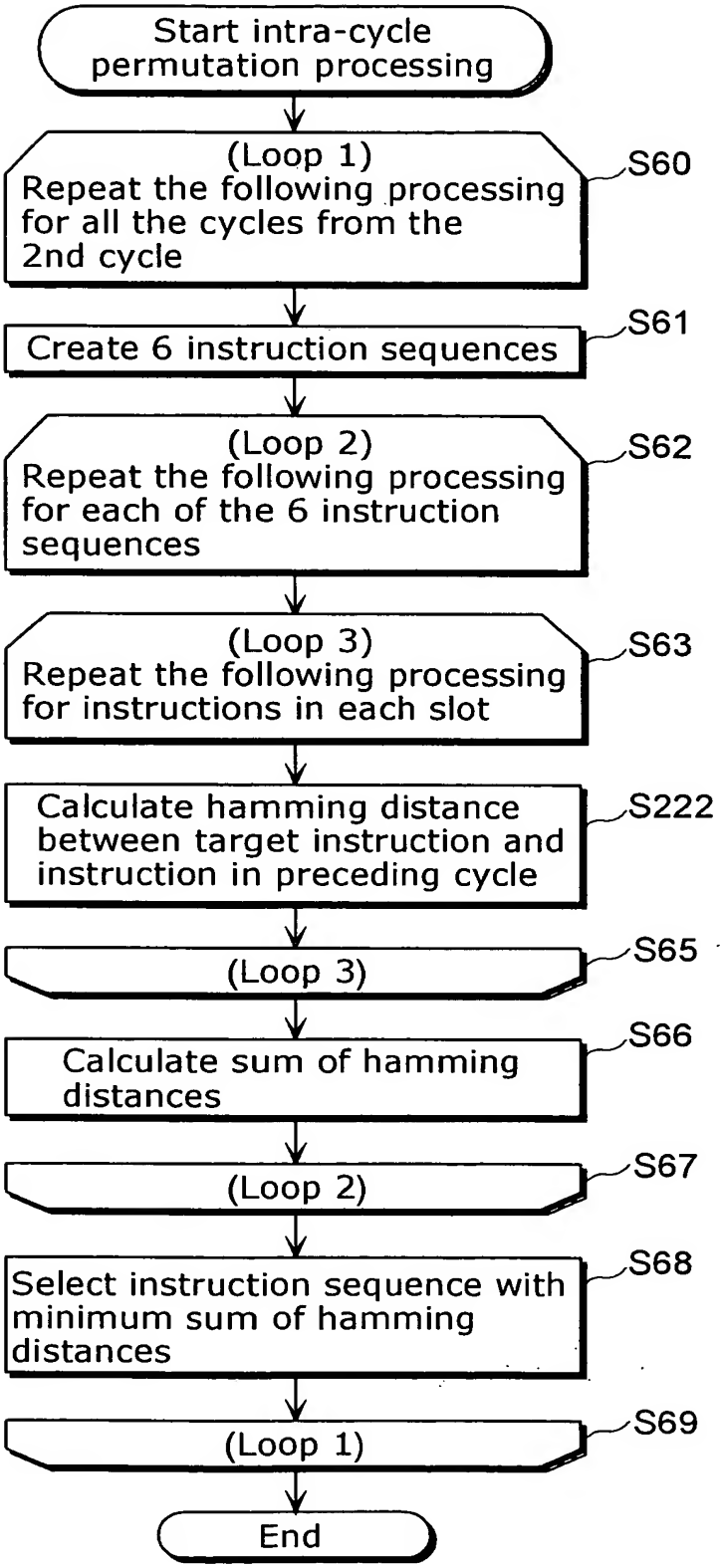

Start intra-cycle permutation processing
(Loop 1) Repeat the following processing for all the cycles from the 2nd cycle
S60
Create 6 instruction sequences
S61
(Loop 2) Repeat the following processing for each of the 6 instruction sequences
S62
(Loop 3) Repeat the following processing for instructions in each slot
S63
Calculate hamming distance between target instruction and instruction in preceding cycle
S222
(Loop 3)
S65
Calculate sum of hamming distances
S66
(Loop 2)
S67
Select instruction sequence with minimum sum of hamming distances
S68
(Loop 1)
S69
End

FIG. 42

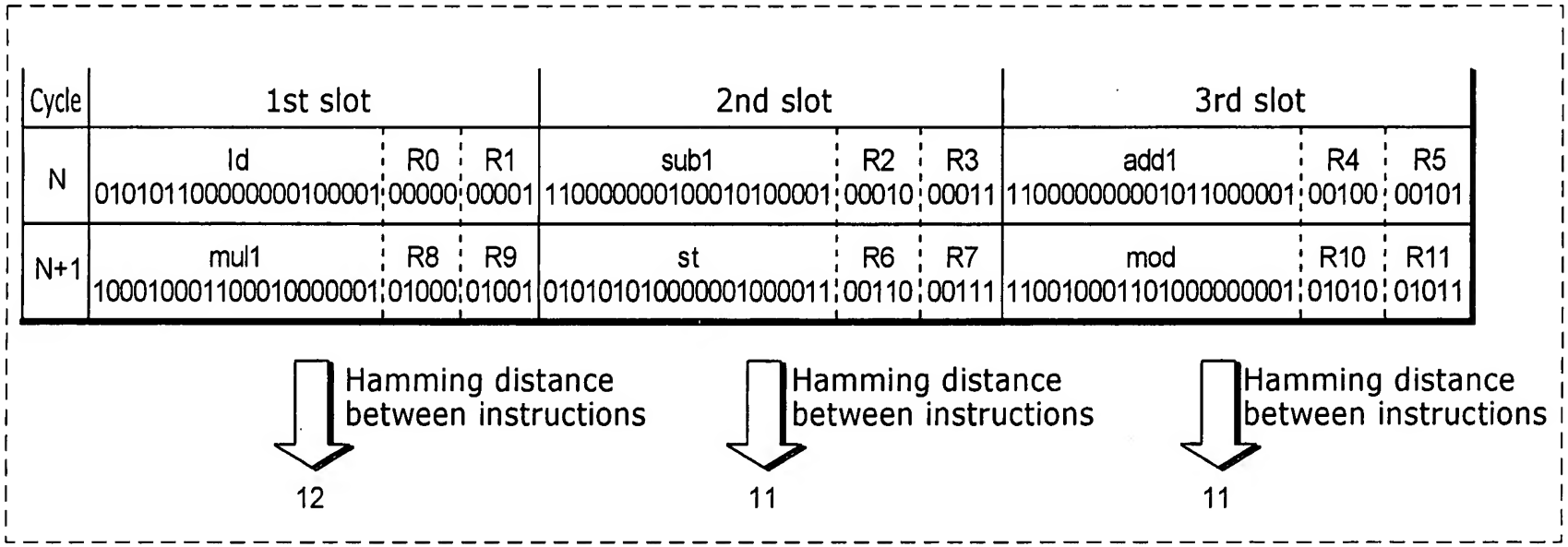

Cycle
1st slot
2nd slot
3rd slot
N
ld
01010110000000100001
R0
00000
R1
00001
sub1
11000000010010100001
R2
00010
R3
00011
add1
11000000000101100001
R4
00100
R5
00101
N+1
mul1
10001000110001000001
R8
01000
R9
01001
st
01010101000001000011
R6
00110
R7
00111
mod
11001000110100000001
R10
01010
R11
01011
Hamming distance between instructions
12
Hamming distance between instructions
11
Hamming distance between instructions
11

FIG. 43

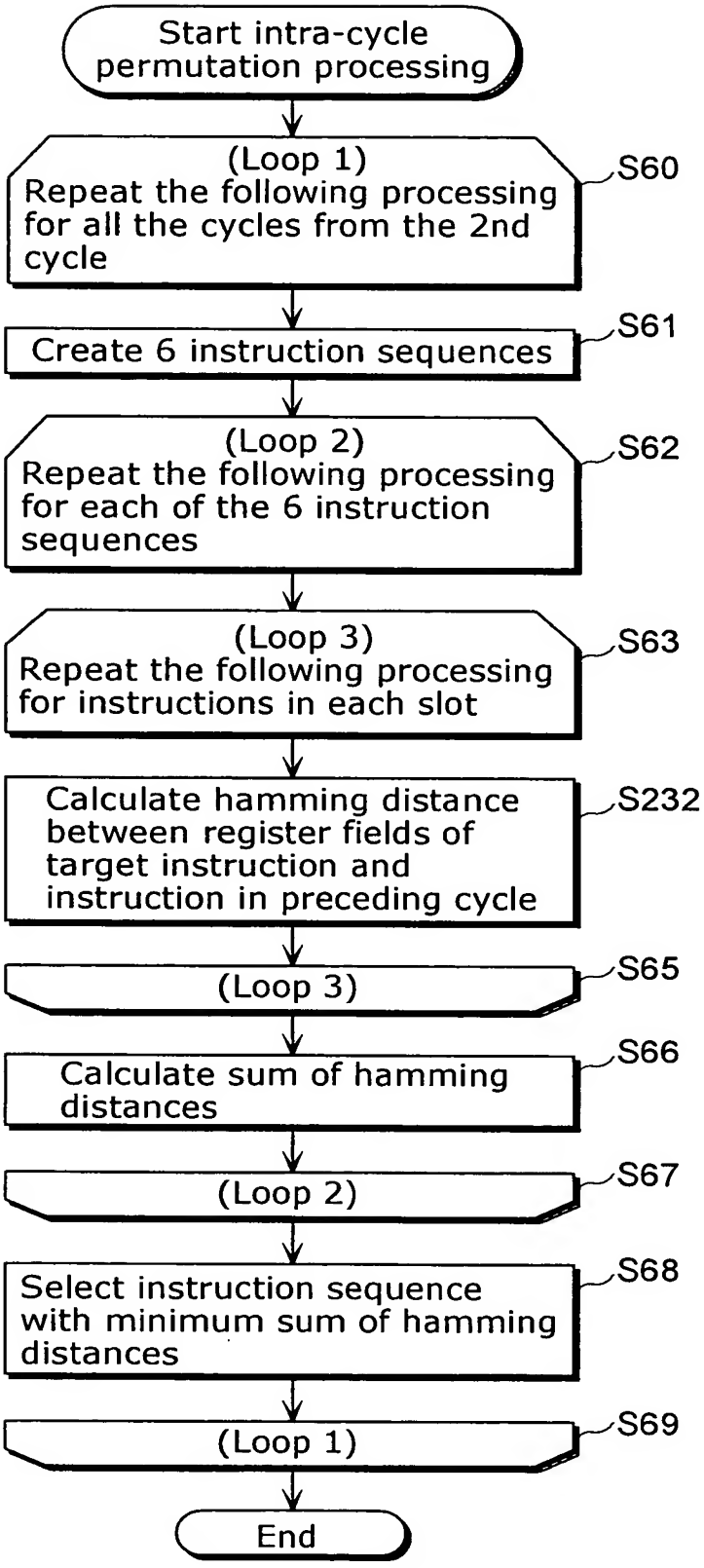

Start intra-cycle permutation processing
(Loop 1)
Repeat the following processing for all the cycles from the 2nd cycle
S60
Create 6 instruction sequences
S61
(Loop 2)
Repeat the following processing for each of the 6 instruction sequences
S62
(Loop 3)
Repeat the following processing for instructions in each slot
S63
Calculate hamming distance between register fields of target instruction and instruction in preceding cycle
S232
(Loop 3)
S65
Calculate sum of hamming distances
S66
(Loop 2)
S67
Select instruction sequence with minimum sum of hamming distances
S68
(Loop 1)
S69
End

FIG. 44

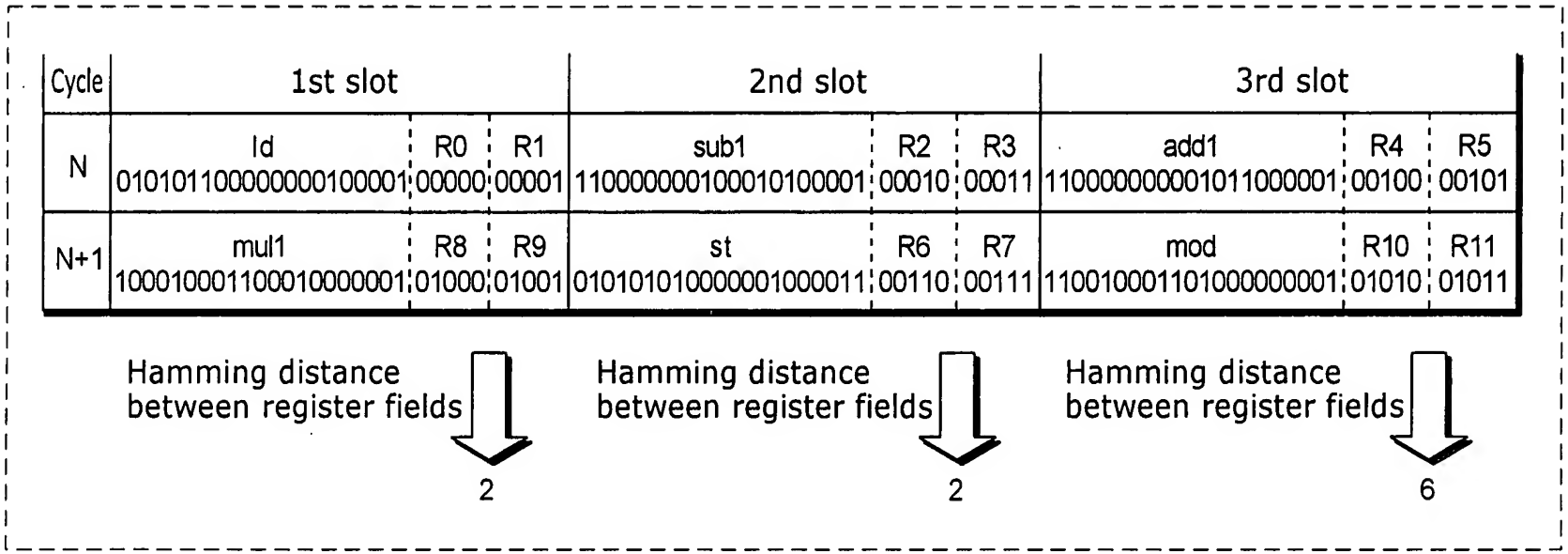

Cycle
1st slot
2nd slot
3rd slot
N
ld
0101011000000000100001
R0
00000
R1
00001
sub1
1100000001000101000001
R2
00010
R3
00011
add1
1100000000001011000001
R4
00100
R5
00101
N+1
mul1
1000100011000100000001
R8
01000
R9
01001
st
0101010100000010000011
R6
00110
R7
00111
mod
1100100011010000000001
R10
01010
R11
01011
Hamming distance between register fields
2
Hamming distance between register fields
2
Hamming distance between register fields
6

FIG. 45

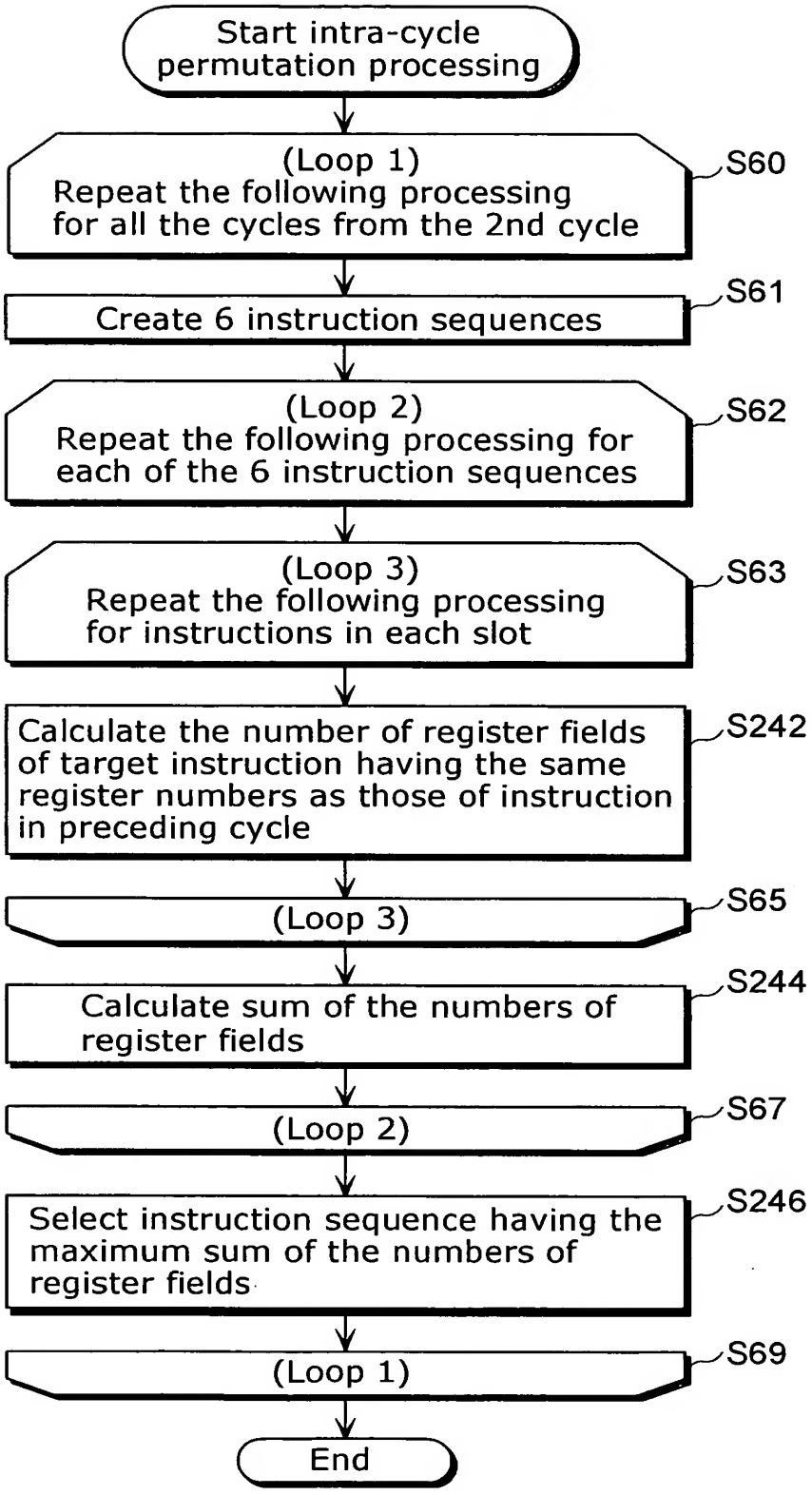

Start intra-cycle permutation processing
(Loop 1) Repeat the following processing for all the cycles from the 2nd cycle
S60
Create 6 instruction sequences
S61
(Loop 2) Repeat the following processing for each of the 6 instruction sequences
S62
(Loop 3) Repeat the following processing for instructions in each slot
S63
Calculate the number of register fields of target instruction having the same register numbers as those of instruction in preceding cycle
S242
(Loop 3)
S65
Calculate sum of the numbers of register fields
S244
(Loop 2)
S67
Select instruction sequence having the maximum sum of the numbers of register fields
S246
(Loop 1)
S69
End

FIG. 46

Cycle	1st slot			2nd slot			3rd slot		
⋮	⋮			⋮			⋮		
N	ld 01010110000000100001	R0 00000	R1 00001	sub1 11000000010010100001	R2 00010	R3 00011	add1 11000000000101100001	R4 00100	R5 00101
N+1	st 01010101000000100011	R5 00101	R8 01000	mul1 10001000110001000001	R2 00010	R3 00011	mod 11001000110100000001	R0 00000	R10 01010
⋮	⋮			⋮			⋮		

FIG. 47A

Cycle	1st slot			2nd slot			3rd slot		
N+1	st 01010101000000100011	R5 00101	R8 01000	mul1 10001000110001000001	R2 00010	R3 00011	mod 11001000110100000001	R0 00000	R10 01010

FIG. 47B

Cycle	1st slot			2nd slot			3rd slot		
N+1	st 01010101000000100011	R5 00101	R8 01000	mod 11001000110100000001	R0 00000	R10 01010	mul1 10001000110001000001	R2 00010	R3 00011

FIG. 47C

Cycle	1st slot			2nd slot			3rd slot		
N+1	mul1 10001000110001000001	R2 00010	R3 00011	st 01010101000000100011	R5 00101	R8 01000	mod 11001000110100000001	R0 00000	R10 01010

FIG. 47D

Cycle	1st slot			2nd slot			3rd slot		
N+1	mul1 10001000110001000001	R2 00010	R3 00011	mod 11001000110100000001	R0 00000	R10 01010	st 01010101000000100011	R5 00101	R8 01000

FIG. 47E

Cycle	1st slot			2nd slot			3rd slot		
N+1	mod 11001000110100000001	R0 00000	R10 01010	st 01010101000000100011	R5 00101	R8 01000	mul1 10001000110001000001	R2 00010	R3 00011

FIG. 47F

Cycle	1st slot			2nd slot			3rd slot		
N+1	mod 11001000110100000001	R0 00000	R10 01010	mul1 10001000110001000001	R2 00010	R3 00011	st 01010101000000100011	R5 00101	R8 01000

FIG. 48

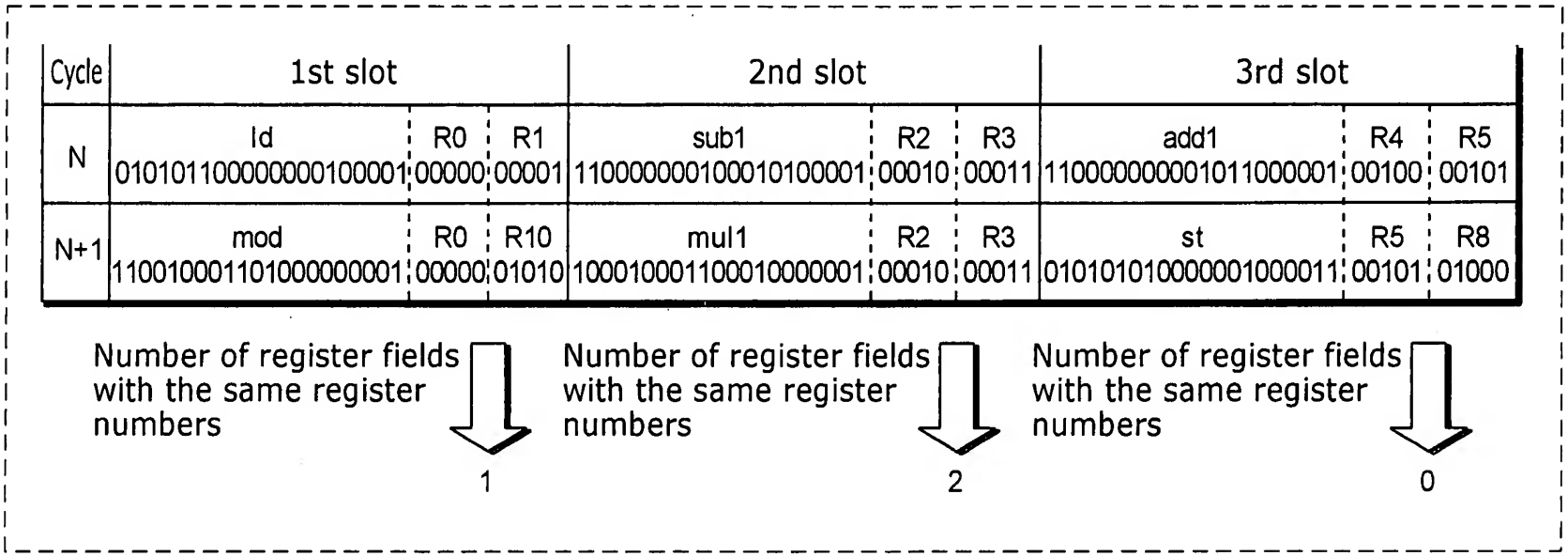
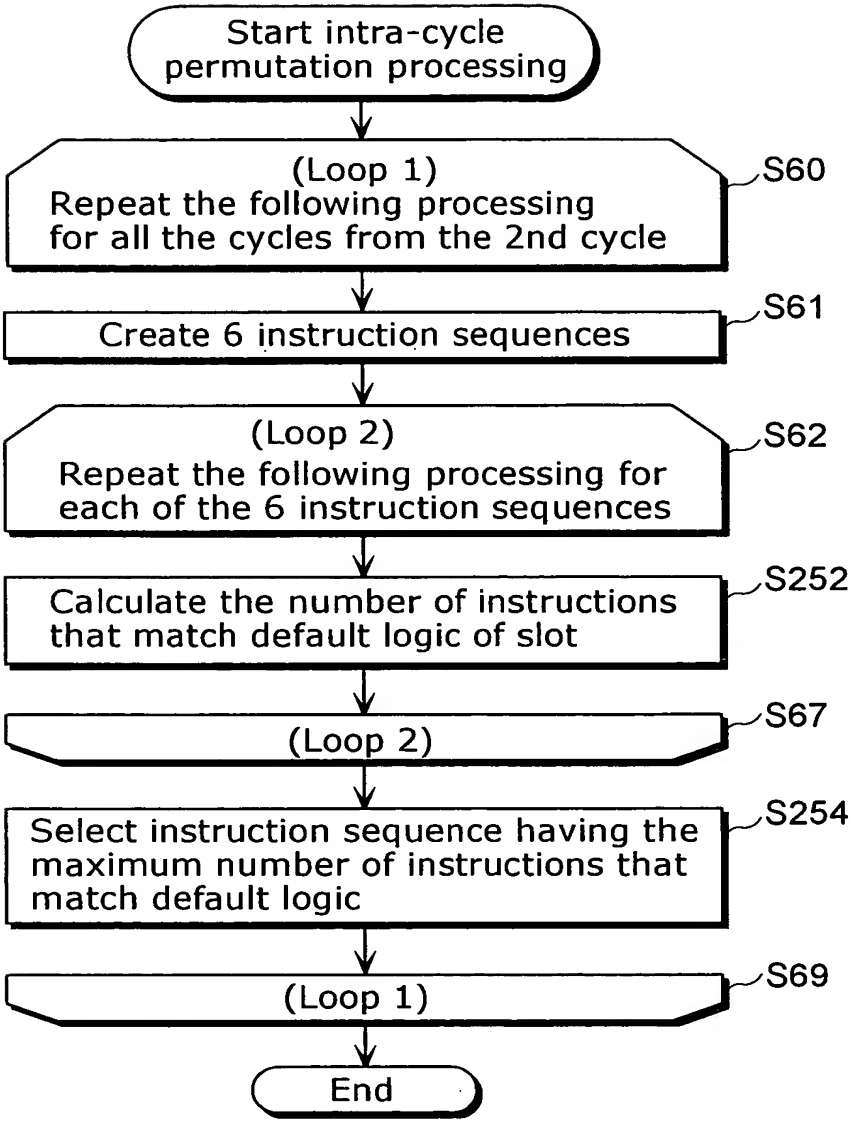

Cycle
1st slot
2nd slot
3rd slot
N
ld
0101011000000000100001
R0
00000
R1
00001
sub1
11000000010001010000 1
R2
00010
R3
00011
add1
11000000000101100000 1
R4
00100
R5
00101
N+1
mod
11001000110100000000 1
R0
00000
R10
01010
mul1
10001000110001000000 1
R2
00010
R3
00011
st
01010101000000100001 1
R5
00101
R8
01000
Number of register fields with the same register numbers
1
Number of register fields with the same register numbers
2
Number of register fields with the same register numbers
0

FIG. 49


Start intra-cycle permutation processing
(Loop 1)
Repeat the following processing for all the cycles from the 2nd cycle
S60
Create 6 instruction sequences
S61
(Loop 2)
Repeat the following processing for each of the 6 instruction sequences
S62
Calculate the number of instructions that match default logic of slot
S252
(Loop 2)
S67
Select instruction sequence having the maximum number of instructions that match default logic
S254
(Loop 1)
S69
End